

SUPER

Anno 2
n° 4 - Aprile 1985

L. 7.500

4

Spedizione in
abbonamento
postale Gruppo III/70

sinc

SUPERSINC
la rivista per gli utenti
dei prodotti SINCLAIR

Una pubblicazione della

J.soft EDITRICE

in collaborazione con

GRUPPO
EDITORIALE
JACKSON



SPECTRUM SHOPPING 2

CORSO DI L/M: 2ª PARTE

SISTEMA SOLARE PER ZX81

MUSIC

SINC 3D PER SPECTRUM

DAL BASIC AL PROLOG



Dalla grande edicola Jackson

Tutto sul personal computer



PERSONAL O

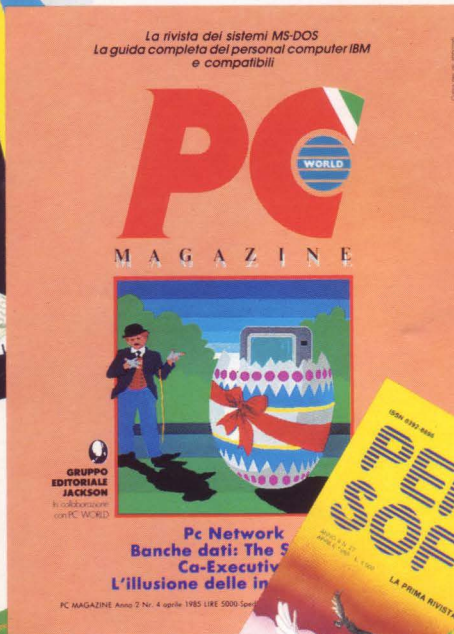
In questo numero:
Sci e videotex

Personal
in salopette,
l'elettrodomestico è salvo

Scheda telematica
su M24

M10 senza veli

Toga per un personal



PC MAGAZINE

In questo numero:
Pc Network
Banche dati: The Source
Ca-Executive
L'illusione delle interfacce ...



BIT

In questo numero:
Speciale Grafica

Bitest:
VDS ECO 3

In prova:
Sharp PC-1260
Copal SC-1200

Softest: Jane



PERSONAL SOFTWARE

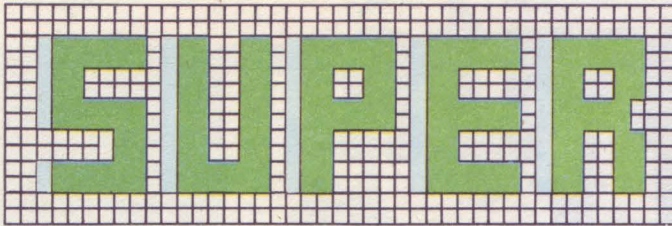
In questo numero:
Equadif per C64
Basket per Spectrum
Battaglia del lago
ghiacciato per C16
Totocalcio per Apple



Personal-O/PC/Bit/Personal Software
sono pubblicazioni firmate:

GRUPPO EDITORIALE JACKSON

via Rosellini, 12-20124 Milano



5 EDITORIALE
POINT

FARE
a cura di **Pietro Dell'Orco**

7 **READ & WRITE**

LA POSTA DEI LETTORI

9 **OPEN**

SPECTRUM SHOPPING 2
trad. e adatt.
di **Marcello Spero**

15 LA CULTURA  INFORMATICA
...then

**DAL BASIC AL PROLOG:
LA BABELLE
ELETTRONICA**
di **Marcello Spero**

19 **REM:HW**

**ESTENDIAMO LE
POSSIBILITÀ DELLA
NOSTRA INTERFACCIA
CON UN CONVERTITORE
ANALOGICO/DIGITALE**
di **Marcello Spero**
Prima parte

23 **RANDOMIZE**

**L'ASSEMBLY
ASSIEME**
di **Marcello Spero**
Seconda parte

31 **ALLACCIAMO LE
STRINGHE**
di **Carlo Panzalis**

39 **LOAD**

39 **GITA TURISTICA**
di **Colin Gooch**
trad. e adatt.
di **Carlo Panzalis**

51 **RACER**
di **David Price**
trad. e adatt.
di **Marcello Spero**

54 **SINC 3D**
di **Martin Clayton**
trad. e adatt.
di **Marcello Spero**

57 **SISTEMA SOLARE**
di **Guido Bertolotti**

62 **TRADUZIONE
NUMERI**
di **Guido Bertolotti**

66 **MUSIC**
di **Giovanni
Benintende**

70 **PESI MOLECOLARI**
di **Antonio Motta**

73 **PRINT**

RECENSIONI LIBRI
di **Lucio Bragagnolo**

75 **ZX CLUB**

**I CLUB
SINCLAIR**

81 **input-output**

PICCOLI ANNUNCI

J. soft s.r.l.
**DIREZIONE, REDAZIONE,
AMMINISTRAZIONE**
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228-68.37.97
68.80.841/2/3

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Lucio Bragagnolo
Marcello Spero

**HANNO COLLABORATO A
QUESTO NUMERO**

Giovanni Benintende
Guido Bertolotti
Antonio Motta
Carlo Panzalis

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonatti

**AUTORIZZAZIONE ALLA
PUBBLICAZIONE:**
Tribunale di Milano n° 199
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e
l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 Milano
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la
DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento
postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500
(11 numeri con cassetta) L. 38.500
(11 numeri senza cassetta) per l'estero:
L. 110.000 (11 numeri con cassetta)
L. 50.000 (11 numeri senza cassetta)
i versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5
20124 MILANO

mediante emissione di assegno
bancario, cartolina, vaglia o
utilizzando il c/c postale
n. 19445204.
Per i cambi di indirizzo indicare,
oltre al nuovo, anche l'indirizzo
precedente ed allegare alla
comunicazione l'importo di L. 500,
anche in francobolli
© TUTTI I DIRITTI DI
RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI
SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLA RIVISTA
ZX COMPUTING - ARGUS
SPECIALIST
PUBLICATION L.T.D.
1 GOLDEN SQUARE LONDON
W 1R3AB



Rivista associata
all'Unione
Stampa
Periodica
Italiana

Novità firmate Jackson.



Giulio Carducci

LE APPLICAZIONI DEL COMPUTER NELL'UFFICIO MODERNO

Questo libro vuole essere un'introduzione all'informatica e, nel contempo, una guida all'utilizzo consapevole dello strumento del giorno, il personal computer, nell'ufficio moderno e nello studio professionale. È rivolto pertanto, ad un vasto pubblico: addetti ai vari settori dell'azienda, quadri, dirigenti, ingegneri, architetti, professionisti in genere.

Cod. 407H Pag. 132 Lire 23.000

Michael Browne

UNITÀ A DISCHI PER MICROCOMPUTER

Il libro, destinato a lettori con una buona conoscenza di base dell'uso di un calcolatore e del linguaggio BASIC, descrive il funzionamento dell'unità a dischi di un personal computer e il significato dei comandi relativi, con particolare attenzione per le diverse tecniche di gestione dei file su disco.

Il libro è arricchito di diversi programmi esempio in BASIC Commodore 4.0, relativi alle diverse tecniche di organizzazione e gestione dei file.

Cod. 300P Pag. 156 Lire 15.000

Mauro Salvemini

URBANISTICA E INFORMATICA

Sempre più vaste sono le applicazioni dell'informatica in ogni settore della vita e del lavoro, e in particolare quelle della computer grafica. Neanche mestieri e professioni con una origine antica come l'architetto o l'urbanista si "salvano" dall'onda dell'informatica.

Un entusiasta utilizzatore di quest'area culturale racconta in questo libro per i colleghi e gli studenti di architettura e di urbanistica tutti i vantaggi e gli avanzamenti possibili nella progettazione e lo studio di nuovi edifici, complessi edilizi o urbani grazie alla computer grafica.

Cod. 801P Pag. 224 Lire 30.000



GRUPPO EDITORIALE JACKSON

La biblioteca che fa testo.

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano
CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione:

- Allego assegno della Banca
- Allego fotocopia del versamento su c/c n. 11666203 a voi intestato
- Allego fotocopia di versamento su vaglia postale a voi intestato

n° _____

Nome _____

Cognome _____

Via _____

Cap _____ Città _____ Prov. _____

Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. _____

ORDINE MINIMO L. 50.000

Graziella Tongoni

LA COMUNICAZIONE CAMBIATA

Questo libro vuole offrire una occasione a tutti i suoi lettori per analizzare, scoprire e potenziare le proprie capacità comunicative e divenire protagonisti reali di una ampia trasformazione in corso, di cui uno degli aspetti più evidenti è costituito dall'espandersi progressivo delle tecnologie.

Cod. 538P Pag. 112 Lire 10.500

Arthur Naiman

WORD STAR

Perché un computer, anche di categoria micro, sia utile nell'attività di ufficio è essenziale che permetta anche di effettuare l'elaborazione dei testi, con tutte le funzioni che ciò comporta. I prodotti software a ciò destinati sono oggi moltissimi ma solo alcuni emergono per completezza e flessibilità: fra questi Wordstar è sicuramente uno dei più riusciti.

Cod. 525P Pag. 222 Lire 23.000

POINT



Fare!

Da sempre l'uomo è in cerca di nuove esperienze. Quella dell'informatica è forse una delle più affascinanti. Ovviamente tale esperienza dev'essere affrontata nel modo giusto.

Oggi tutti si sforzano di dare consigli su come muoversi nell'apparentemente intricato mondo dei calcolatori. Non è necessario essere esperti matematici oppure piccoli geni per avvicinare la materia.

Fortunatamente c'è chi, come sempre, pensa e provvede anche per chi non ha mezzi o capacità di fare in proprio. Ecco dunque spiegata la nascita di una miriade di pubblicazioni, la disponibilità di centinaia di programmi pronti all'uso e via dicendo.

Forse il discorso è banale, ma provando a riflettere si constaterà come tutto ciò abbia contribuito in maniera determinante alla diffusione dei personal e home computer. Si sa, l'hardware senza il software non ha storia. Il problema, per l'utente medio, è quello di non fossilizzarsi solo perché c'è comunque qualcuno che ci pensa. La miglior politica è quella di considerare tutto come un aiuto per imparare a far da sé. Solo in questo modo si potrà procedere, magari a piccoli passi, fino a padroneggiare in qualche modo la situazione.

Ogni nuova scoperta deve servire da pretesto per cimentarsi; aspettare che la soluzione di un problema specifico arrivi dalle pagine di una rivista o dalla commercializzazione di un nuovo prodotto non è certo la strada giusta.

L'attesa passiva non farà vivere l'esperienza con il computer nel modo corretto.

Dunque, sotto! Ogni riga di programma dev'essere uno spunto, la routine "wyz" deve servire a risolvere quel problema che vi assilla da giorni.

La parola d'ordine è quindi "fare", poi, per quelli che riescono meglio, c'è la possibilità di inviare i propri lavori in redazione per l'eventuale pubblicazione sulla rivista che, oltre alla soddisfazione, può far guadagnare qualche lira.

Pietro Dell'Orco

Piccola guida per l'input dei programmi

I programmi pubblicati da SUPERSINC sono stati accuratamente provati e verificati: in questo modo speriamo di avere ridotto al minimo la possibilità di errori. Nel caso comunque un programma dovesse presentare malfunzionamenti, pubblicheremo le modifiche da approntare in uno dei numeri immediatamente successivi della rivista.

Per utilizzare i programmi pubblicati è sufficiente digitare i relativi listati sulla tastiera del calcolatore.

Nel caso dello ZX SPECTRUM, abbiamo previsto una semplice decodifica dei tasti da digitare nel modo grafico, per semplificare la comprensione dei listati.

Come è noto, lo SPECTRUM è provvisto di 2 serie di tasti grafici: una prima serie di 16 caratteri grafici predefiniti (i tasti numerici da 1 a 8 e gli stessi "shiftati") e una serie di caratteri definibili dall'utente nell'ambito di un programma (le lettere da A a U).

In entrambi i casi per ottenere i caratteri desiderati occorre entrare in modo grafico (cursore contrassegnato dalla lettera G lampeggiante), premendo contemporaneamente i tasti CAPS-SHIFT e 9.

Nei nostri listati i caratteri grafici predefiniti sono indicati da una Q seguita dal numero corrispondente al testo che occorre digitare, il tutto racchiuso tra due parentesi graffe.

Ad esempio {G4} significa che occorre digitare il tasto 4, con il cursore in modo grafico.

Analogamente la codifica SG, seguita da un numero da 1 a 8, significa che occorre digitare il relativo tasto numerico, premendo contemporaneamente il tasto CAPS-SHIFT.

Ad esempio quando, si trova la codifica {G2} occorre premere il tasto 2 contemporaneamente al tasto CAPS-SHIFT, ovviamente con il cursore in modo grafico. In entrambi i casi precedenti, quando un simbolo grafico deve essere digitato più volte, i caratteri G o SG della codifica sono preceduti da un numero che specifica quante volte va premuto il tasto grafico indicato.

Così ad esempio {8G5} significa che il tasto grafico 5 va digitato 8 volte e analogamente {4SG1} indica che il tasto grafico 1 e CAPS-SHIFT devono essere battuti 4 volte.

I caratteri grafici definibili (le lettere da A a U in modo grafico) hanno una codifica semplificata: la lettera corrispondente, sottolineata.

Quando in un listato viene presentata, ad esempio, una A sottolineata occorre entrare in modo grafico (al solito premendo contemporaneamente i tasti CAPS-SHIFT e 9) e quindi digitare semplicemente il tasto che contrassegna la lettera A.

Quando leggete	Premete	Vedrete
{G1}		
{G2}		
{G3}		
{G4}		
{G5}		
{G6}		
{G7}		
{G8}		
{SG1}		
{SG2}		
{SG3}		
{SG4}		
{SG5}		
{SG6}		
{SG7}		
{SG8}		

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

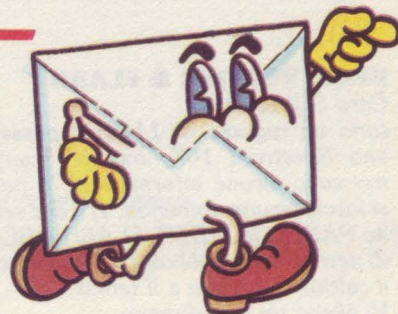
Quando leggete	Premete	Vedrete
<u>A</u>		Simbolo grafico definito nel programma in uso.
<u>B</u>		
<u>C</u>		
<u>D</u>		
<u>E</u>		
<u>F</u>		
<u>G</u>		
<u>H</u>		
<u>I</u>		
<u>J</u>		
<u>K</u>		
<u>L</u>		
<u>M</u>		
<u>N</u>		
<u>O</u>		
<u>P</u>		
<u>Q</u>		
<u>R</u>		
<u>S</u>		
<u>T</u>		
<u>U</u>		

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

READ &WRITE

La posta
dei lettori



Grafici di carattere

Cara redazione,
non intendo spendere tempo e carta per le lodi alla vostra rivista, e passo subito al nocciolo della questione: come mi devo comportare quando leggo, nei listati per lo Spectrum, quelle benedette lettere sottolineate?

Riccardo Di Nuzzo - Rimini

Caro Riccardo, quelle "benedette" lettere vanno inserite in modo grafico; ossia, prima di batterle devi premere insieme CAPS SHIFT e il tasto 9, accertandoti che il cursore diventi una "G" lampeggiante. A questo punto puoi inserire le lettere sottolineate, senza curarti della sottolineatura, che è solo una speciale codifica adottata per distinguere i caratteri grafici da quelli alfabetici.

Completato l'inserimento? Bene: schiaccia il tasto 9, assicurati che il cursore sia nuovamente rappresentato dalle solite "K", "L" o "C" e prosegui tranquillamente. E per questa volta passi: ma la prossima non saremo così buoni, e ti consiglieremo semplicemente di essere meno pigro. Non ti sei accorto che sia su SUPERSINC che su PAPERSOFT c'è sempre una pagina chiamata "guida all'input dei programmi"?

Ancora sullo ZX 81

Non comprendo il motivo per cui il software per ZX 81 è inferiore (come numero) a quello per Spectrum. Trovo che questa differenza non sia giusta...

Mario Artuso - Bessica (TV)

Ci ripetiamo per l'ennesima volta: non siamo contrari ad aumentare lo spazio dedicato allo ZX81 e, compatibilmente con i suoi problemi di affidabilità (la presenza sul numero di febbraio di un solo programma per ZX81 è dovuta sostanzialmente a questo), se riterremo esserci una base di utenti ZX81 che giustifichi il nostro sforzo, lo faremo senz'altro. Anche se la richiesta di dividere in due la rivista ci pare un po' eccessiva...

In cerca di guerre

Gentilissima redazione, gradirei poteste indicarmi se in commercio vi sono programmi di guerra simula-

ta.

Sono già in possesso di "Stonkers", e lo apprezzo moltissimo...

**Giuseppe Rimondotto
Orbassano (TO)**

Dopo aver solleticato il nostro database ne abbiamo ricavato alcuni titoli. La lista che le presentiamo comprende non solo ciò che lei cerca, ma anche giochi interessanti e definibili più genericamente "strategici", un esempio dei quali è "Hunter Killer" (che - non lo dica a nessuno - le consigliamo). Naturalmente non si tratta di un elenco completo, né estremamente aggiornato - i programmi per Spectrum spuntano come i funghi - né, soprattutto, critico (o non basterebbe un intero numero della rivista). Ecco i titoli: Airline, Angler, Apocalypse, Battle 1917, Battle of Britain, Brewery, British Lowland, Conflict, Confrontation, Conquest, Dix Mille, Doomdark's Revenge (non propriamente definibile come simulazione strategica, ma di estrema complessità), Fall of Rome, Galaxy Conflict, Jonny Reb, Lords of Midnight (vedasi "Doomdark's Revenge"), Paratroopers, Plunder, Rebelstar Raiders, Red Weed, Reichswald, Roman Empire, Sheepwalk, Viking Raiders, War 70, Wilfred the Hairy.

Crediamo che sia abbastanza, se non altro per pensarci su e cominciare a informarsi in merito...

I joystick, questi sconosciuti

Spettabile redazione, desidererei alcune informazioni circa i joystick e le relative interfacce:

1) qual è la differenza tra un'interfaccia programmabile e una non programmabile?

2) con una di queste due interfacce si possono utilizzare tutti i giochi presenti sul mercato o no?

3) in che cosa consiste la programmazione dell'interfaccia? È molto difficile?

4) i joystick sono compatibili con tutte le interfacce o bisogna tener conto di marche e modelli ben precisi?

Vi ringrazio anticipatamente.

**Alberto Okely
Cernusco sul Naviglio (MI)**

Le domande sono un po' ambigue, ma cercheremo di essere più chiari possibili.

Dunque:

1) Il fatto di poter essere programmata o meno! Scherzi a parte, possiamo vedere i joystick come dei sostituti della tastiera, più comodi ed efficienti.

Un'interfaccia programmabile consente all'utente la scelta dei tasti da sostituire, mentre una non programmabile richiede, per funzionare, che il programma preveda l'uso del joystick in questione, e quindi contenga il software occorrente.

2) Con un'interfaccia programmabile, in teoria, sì: con una non programmabile, c'è compatibilità solo con i programmi che la prevedono. In pratica, da un lato possono esserci isolate eccezioni, mentre dall'altro virtualmente tutti i programmi più diffusi sono compatibili con i joystick più famosi (qualche nome? Kempston, Fuller, Protek, Sinclair...). Nel caso delle interfacce programmabili è importantissimo che il software di controllo dell'interfaccia non risieda nella RAM dello Spectrum occupata dal programma, pena conseguenze facilmente immaginabili.

3) La programmazione dell'interfaccia, in parole poverissime, consiste nel dire al computer di quali tasti il joystick farà le veci. Le modalità concrete della programmazione variano moltissimo da un'interfaccia all'altra, ma il peggio che può capitare è di dover battere qualche riga in BASIC, magari aiutati dalle istru-

zioni allegate — si spera — all'interfaccia.

4) Un joystick in grado di collegarsi materialmente a un'interfaccia è, generalmente, compatibile con essa. Possono però verificarsi eccezioni, che dipendono dal meccanismo di funzionamento interno del joystick e vanno esaminate caso per caso.

BEEP, BORDER & FLASH

Caro SUPERSINC,

sono un ragazzo di 11 anni, posseggo uno Spectrum 16K, ma nel manuale mancano alcune informazioni, e perciò vi faccio alcune domande: 1) a che servono PAPER, BORDER, FLASH e BEEP? 2) perché non pubblicate dei giochi come il calcio, il pugilato e il tennis? Vi allego un mio disegno.

Roberto Ortolano

Caro Roberto, PAPER modifica il colore dello schermo, della "carta" su cui scrivi i tuoi programmi. Esempio: accendi il tuo Spectrum e digita PAPER 1: poi premi due volte ENTER e vedrai lo schermo diventare blu. BORDER cambia il colore del bordo dello schermo, e come esempio puoi digitare BORDER 2, sempre seguito da ENTER.

Ora FLASH: questo comando fa lampeggiare una scritta. Prova a digitare PRINT FLASH 1; "Guarda come lampeggio!!!": PRINT "Ora non lampeggio più...", più ENTER. Infine, digitando BEEP 1, 1 (seguito come al solito da ENTER) potrai sentire il tuo Spectrum suonare! Contento? Ancora una cosa: il tuo disegno è bellissimo, ma è troppo piccolo e troppo poco colorato; se ce lo rispedirai con le caratteristiche necessarie, lo pubblicheremo, OK?

Exotic shopping

Spettabile redazione di SUPERSINC, vorrei aver informazioni riguardanti l'acquisto di materiale in Inghilterra.

Avevo scritto una prima lettera simile a questa, ma credo che per forza maggiore voi non abbiate potuto soddisfare le mie esigenze.

Ma ora passiamo agli affari; io sarei interessato all'acquisto dei Microdrive e dell'Interfaccia 1, e vorrei sapere come fare per acquistarli in Inghilterra. Vi prego di soddisfare il mio desiderio al più presto, visto che i Microdrive mi occorrono con una certa urgenza.

**Andrea Zavaglia
Gambettola (FO)**

Tutte le informazioni utili sono apparse in "Spectrum Shopping", apparso nei numeri di SUPERSINC datati luglio/agosto e settembre 1984, nonché nel numero scorso (e in questo).

Parliamo ancora di joystick

Sono un entusiasta utente di ZX Spectrum, e non proprio un neofita, che ha deciso di munirsi di un joystick.

Visto che per questo ho bisogno di una interfaccia, è mio desiderio comprarne una buona.

Scarto subito la ZX Interface 2, che con i miei giochi non ha molto da spartire (niente di personale, comunque).

In un articolo da me letto ho visto notizia dell'interfaccia Kempston, che, secondo l'articolo, sarebbe perfino migliore dell'interfaccia 2. Ma è vero?

Anche quest'ultima ha la possibilità di utilizzare il nuovo software su cartuccia; ma ne esiste a sufficienza, tanto da giustificare la spesa per l'interfaccia?

Sperando che i miei quesiti possano avere risposta, vi ringrazio per il vostro interessamento.

Fabiano Cassioli - Firenze

Siamo tutto sommato d'accordo nel preferire l'interfaccia Kempston alla ZX Interface 2, tra l'altro più costosa. Ci piace meno il discorso sulle cartucce ROM, su cui riteniamo opportuno fare un paio di precisazioni. Innanzitutto il software esistente su cartuccia ROM non è affatto "nuovo", ma si compone per lo più di grandi classici, come "Jetpac". In secondo luogo le cartucce, oltre che poche e poco diffuse, sono anche costose, al punto da far dubitare della convenienza del loro utilizzo per quelli che, tutto sommato, sono giochi e non programmi commerciali. Una buona soluzione, a questo punto, potrebbe essere l'acquisto di una interfaccia joystick compatibile Kempston, rinunciando all'idea delle cartucce ROM. O no?

Grazie, ma...

Vi propongo con la presente una nuova applicazione per lo Spectrum: un programma che [...]. Vi prego quindi di considerare la mia proposta di pubblicare un programma di questo tipo, magari corredandolo di un opuscolo che ne illustri dettagliatamente gli aspetti.

Se la questione vi interessa, vi invito a contattarmi in modo assolutamente non impegnativo, per ora, per nessuna delle due parti [...].

Francesco Santi - Bologna

Egregio signor Santi, siamo letteralmente sommersi da programmi di tutti i generi inviati dai lettori, e non abbiamo assolutamente la possibilità (oltre che la volontà) di intraprendere trattative su materiale di cui non siamo neanche in possesso. La invitiamo piuttosto a mandarci il suo programma: se dovesse risultare interessante, ci metteremo in contatto con lei al più presto.

Attenzione!

Alcuni dei programmi presenti su questo numero di Supersinc si avvolgono di particolari tecniche di programmazione, che possono provocare inconvenienti durante il caricamento del programma ad essi successivo.

Per evitare problemi, vi consigliamo di effettuare un reset (spegnendo e riaccendendo il computer o digitando RUN USR O o per per chi ha uno Spectrum Plus, azionando l'apposito pulsante) prima di ogni LOAD.



OPEN

trad. e adatt.
di **Marcello Spero**

Spectrum Shopping 2

La seconda puntata del nostro viaggio tra le periferiche per i computer Sinclair.

STAMPANTI E INTERFACCE PER STAMPANTI

Con la progressiva scomparsa della ZX Printer ci troviamo di fronte alla necessità di una scelta. La ricerca di un diverso mezzo per stampare è sempre stata comunque viva fra gli utilizzatori dei computer Sinclair che sfortunatamente si sono sempre trovati alle prese con elevati prezzi d'acquisto.

Il modo più semplice, economico e comunque perfettamente soddisfacente di procurarsi una stampante è l'acquisto di una Alphacom 32, per circa 69.95 sterline. Completamente compatibile sia con lo ZX81 che con lo Spectrum è fornita con carta termica bianca, che produce caratteri blu non molto intensi. Esiste comunque in commercio carta termica che produce ottimi caratteri neri.

Se la prospettiva di possedere una Alphacom non vi soddisfa, dovrete fare un deciso salto ed indirizzarvi verso il dispendioso mercato delle stampanti a matrice, a margherita, ecc. I modelli qui sono tantissimi, ciascuno con differenti caratteristiche, vantaggi e svantaggi. Un suggerimento è quello di visitare alcuni negozi, naturalmente dopo aver accertato le vostre reali necessità.

Interessante è la Shinwa CP80; con lo Spectrum produce risultati perfetti, mentre purtroppo non è in grado di riprodurre i caratteri grafici dello ZX81. La Epson FX80 è una macchina magnifica, che dovrebbe costare circa 400 sterline. È in grado di produrre listati perfetti con lo ZX81, ma purtroppo distorce la grafica dello Spectrum!

Prima che le rispettive ditte scrivano per protestare, c'è da dire comunque

che è di vitale importanza l'interfaccia che collega il computer alla stampante, e le prestazioni di quest'ultima possono esserne molto influenzate.

Il tipo più comune di collegamento pare essere il Centronics, di cui è dotata la gran parte delle stampanti; il suo rivale, l'RS232, può comunque essere aggiunto, con un costo non elevato.

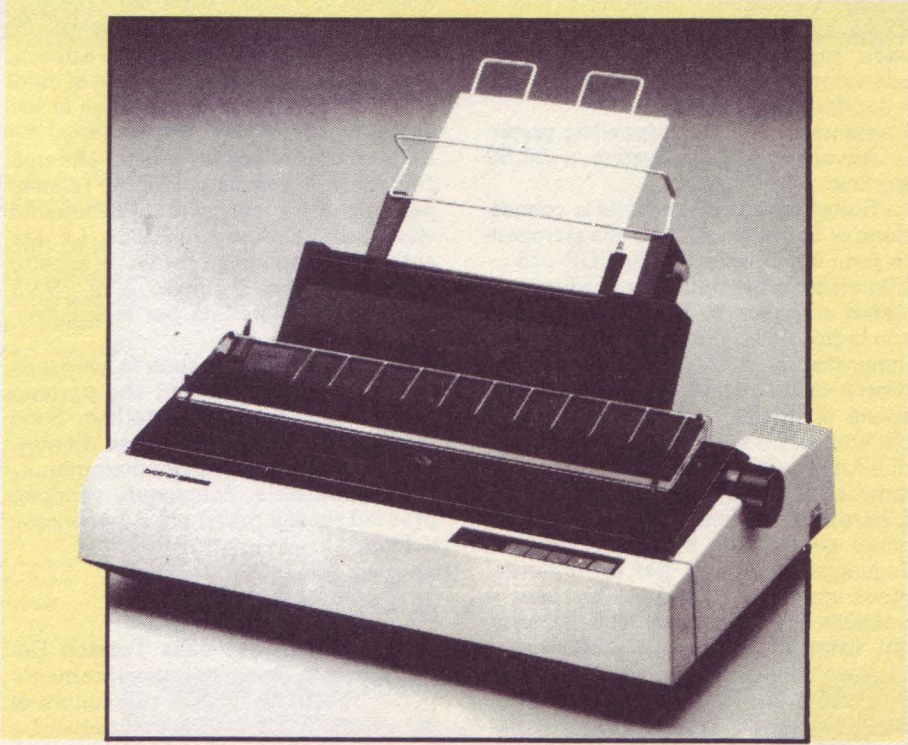
Alcune delle interfacce di più recente produzione sono: DK'Tronics (Centronics) a 39.75 sterline, compresi cavi, software, ecc.

Kempston E, che possiede una ROM incorporata per eliminare la necessità di software per il suo funzionamento. È facile da usare e per questo la raccomandiamo vivamente.

ZXLprint III (34.95 sterline) dispone anch'essa di ROM incorporata, ed ha il vantaggio di essere anche RS232 compatibile. Davvero notevole.

In breve

L'Alphacom 32 viene importata dall'America, dove è usata con la versione Timex dello ZX81. A differenza della stampante Sinclair, essa utilizza un suo alimentatore separato, incluso nella confezione. Accetta tutti i comandi Sinclair e stampa in blu su carta bianca caratteri leggermente più stretti ed alti degli originali. La carta è piuttosto economica: 1 sterlina per un rotolo di 25 metri.



La Hilderbay produce un'interfaccia secondo lo standard Centronics per lo Spectrum ed il software che ne permette il funzionamento. Il suo prezzo, comprensivo di 1 metro di cavo, è di 45 sterline. Il software per l'interfaccia Hilderbay è scritto sia in BASIC che in linguaggio macchina e consente l'uso di LLIST ed LPRINT, oltre ad includere una routine per produrre hardcopy dello schermo. Non manca un mini word-processor prodotto dalla Tasword. Della Cobra Technology sono due interfacce per lo ZX81, che consentono il collegamento rispettivamente a stampanti Centronics o RS232. Si inseriscono al connettore posteriore dello ZX81 e dispongono di baud rate variabile sotto controllo software. Il prezzo di entrambe è di 26 sterline più VAT (la solita IVA inglese).

La Morex Peripherals ha unito in una singola confezione, al prezzo di 39.95 sterline, due interfacce, prodotte in versione per lo ZX81 o per lo Spectrum. Si tratta di un'interfaccia parallela tipo Centronics e di una RS232 bidirezionale. Con questi due congegni diventa possibile la stampa di linee della massima lunghezza consentita dalla stampante, pur usando i comandi BASIC LLIST e LPRINT; il baud rate è variabile. La versione per lo Spectrum utilizza un sistema operativo incorporato, che consente l'uso di package per il word processing.

Un'interfaccia Centronics adatta sia allo ZX81 che allo Spectrum è stata presentata dalla Advanced Digital Systems.

Completa di cavo e connettore Centronics comprende anche software in grado di riconoscere LLIST e LPRINT, consentendo stampe dirette dal BASIC. È garantita per dodici mesi.

In aggiunta alla sua compatibilità con lo ZX81 e Spectrum, l'acquisto di software suppletivo e di un economico adattatore ne consente l'uso anche con il Jupiter Ace.

Il sistema completo è disponibile presso la Advanced Digital System a 34.50 sterline.

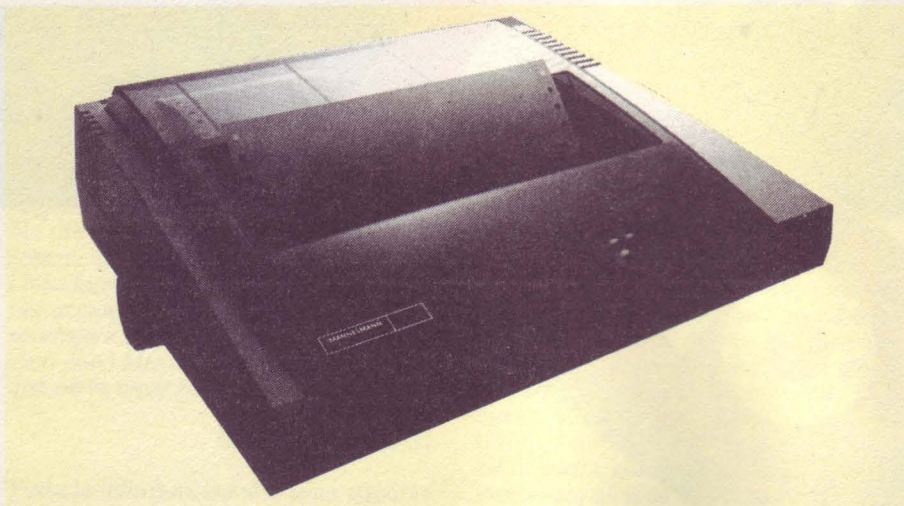
La Softest interface consente la connessione di Spectrum e ZX81 alla stampante semi intelligente Tandy CGP-115.

L'interfaccia permette di riprodurre caratteri e grafica su questa stampante, con la possibilità di scegliere fra diverse dimensioni di stampa e di sovrapporre diversi colori. La stampante infatti dispone di quattro colori: rosso, verde, blu e nero. La sua velocità di stampa è di 52 mm/sec in senso orizzontale e 73 mm/sec in senso verticale.

L'hardware dell'interfaccia è accompagnato dal relativo software, scritto in linguaggio macchina, per controllare il movimento delle penne. Con esso è possibile stampare testi, tabelle, disegni, listati di programmi, diagrammi e titoli a grandi dimensioni.

Il prezzo è di 35 sterline.

Ben disegnata e di facile inserimento è



la Memotech RS232, che possiede software su ROM. Finita nel blu e argento della Memotech, funziona egregiamente. 39.95 sterline.

L'interfaccia Centronics della Memotech assomiglia molto nell'estetica alla versione RS232 e funziona altrettanto bene. 24.95 sterline.

La Sinclair ZX Interface 1 si avvita al di sotto dello Spectrum, inclinandolo in avanti. Può controllare fino a otto microdrive, dispone di una interfaccia RS232 per stampanti di qualsiasi tipo, e può collegare in rete locale fino a 64 Spectrum alla velocità di 100K baud. 50 sterline.

CONSOLE E CONTENITORI

Dover riporre ogni volta tutto l'equipaggiamento è senz'altro seccante, e quindi il primo acquisto è di solito un televisore riservato.

Il problema successivo è come e dove tenere tutto quanto! C'è persino chi ha ampliato la propria casa per creare lo spazio necessario. Non è comunque indispensabile arrivare a questi estremi, e per questo ecco una selezione di modi più comodi per tenere ogni cosa al suo posto.

La Marcol produce un cabinet che può contenere il computer con le relative periferiche e il televisore; la stampante viene posta al di sopra di esso. Le due ante di cui dispone gli danno un aspetto molto elegante. Il prezzo è di 79.95 sterline, più 5 sterline per la spedizione.

Sempre dalla Marcol sono le più tradizionali scrivanie, a prezzi che partono dalle 39.99 fino a 185 sterline. Sono reperibili al Solent Business Center, Millbrook Road West, Southampton.

Molte altre ditte, comunque, vendono prodotti simili a prezzi più o meno equivalenti.

In breve

La Executive Case della Treetop Designs è stata disegnata specificamente per lo Spectrum. Questo contenitore di plastica ABS combina le caratteristiche

di una valigetta ventiquattrore, che vi permette di portare in giro il vostro equipaggiamento, e di una console su cui poter lavorare.

La console dispone di un piano sollevato ed inclinato per lo Spectrum e una nicchia per l'alimentatore; la circolazione dell'aria è assicurata. C'è inoltre spazio per stampante, registratore e per i vari cavi. Non mancano interruttore acceso/spento e LOAD/SAVE.

Sei cassette e un rotolo di carta per la stampante possono trovare posto nella console, ed il coperchio comprende, oltre allo spazio per riporre i cavi, un soffice cuscino di gommapiuma per tener bloccato l'equipaggiamento durante il trasporto. Il suo prezzo è di 47.45 sterline.

Della Computer Cases è il contenitore per Spectrum e ZX81. Si tratta di una valigetta che può contenere lo Spectrum o lo ZX81, l'alimentatore, la stampante, il registratore e delle cassette, ecc. Non manca lo spazio per i cavi. Ogni cosa è protetta con gommapiuma, che può essere tagliata per creare lo spazio ad altre periferiche, come i microdrive. Anch'essa in stile ventiquattrore, nera con finiture metalliche, costa 38.95 sterline.

Una stazione di lavoro per lo Spectrum, costruita in solida plastica ABS, è disponibile presso Peter Furling Products a 13 sterline. Tutti i cavi sono accuratamente nascosti e c'è anche spazio per il televisore. Interruttore LOAD/SAVE, altoparlante e base in metallo si possono avere per un modesto extra.

Fra le console più economiche va annoverata quella in cartone prodotta dalla Print'n' Plotter Products. Venduta al prezzo di 8.25 sterline dà spazio a Spectrum, alimentatore e stampante. Una volta acquistata va montata, ma la cosa è piuttosto semplice con l'aiuto delle istruzioni fornite. Sebbene non ci sia spazio per le periferiche, eventuali modifiche possono essere fatte facilmente. La Desk Console della Traffic Technology è una console da tavolo costruita in grosso ABS nero, con coperchio staccabile e piedini antisdrucciolo.

C'è spazio per Spectrum, alimentatore,

CONVERTIBLE BY FORD.



1300-1600-1600i

Un modo di essere. Il piacere di toccare il mondo. Un desiderio che diventa automobile.

Escort Cabriolet. Quattro comodi posti, un vero bagagliaio che offre una capienza ancora superiore, grazie alla possibilità di reclinare i sedili posteriori.

Escort Cabriolet è soluzioni tecnico-costruttive d'avanguardia: roll-bar integrale, capote completamente foderata e insonorizzata, ripiegabile facilmente senza il minimo sforzo. Lunotto posteriore in vetro riscaldato elettricamente.

Escort Cabriolet è straordinarie prestazioni. Come nella versione 1.6 injection: 186 km/h, da 0 a 100 km/h in 9.9 sec.

Escort Cabriolet. Il piacere dell'automobile allo stato puro.

Prestazioni e consumi	1296 cc	1597 cc	1597 cc-i
Velocità massima	157 km/h	169 km/h	186 km/h
Da 0 a 100 km/h	14.0 sec	11.8 sec	9.9 sec
Consumo a 90 km/h	17.8 km/lt	18.2 km/lt	16.7 km/lt

Tutte le vetture Ford sono coperte da garanzia 1-3-6 (un anno di garanzia estensibile a tre con "La Lunga Protezione" e sei anni di garanzia contro la corrosione perforante) e assistite in oltre 1000 punti di servizio. Finanziamenti Ford Credit e cessioni in Leasing.

ESCORT

Tecnologia e temperamento.

Per collaborare a SUPERSINC

La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair, di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Horance Goes.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e sopra-

tutto le eventuali espansioni richieste. Come suggerimenti di programmazione si consiglia di usare, per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE, piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24x36, o 6x6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il materiale non pubblicato non verrà restituito.

15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviare idee e programmi a:
SUPERSINC
Viale Restelli, 5
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

stampante, interfaccia RS232, interfaccia per joystick, due microdrive, registratore a cassette, cassette. È previsto anche un interruttore, il che vuol dire che non c'è bisogno di staccare i cavi durante LOAD/SAVE. Il suo prezzo è di 42.18 sterline. La Microcase della Micro Aids non è che uno dei numerosi prodotti di questa ditta.

Disegnata per essere compatibile con lo Spectrum e con lo ZX81, è una valigetta di tipo ventiquattrore con coperchio staccabile. L'interno è riempito con gommapiuma per assicurare l'equipaggiamento; può contenere computer, registratore e cassette, alimentatore, stampante e cavi. Il suo prezzo è di 32.95 sterline.

Dalla stessa ditta è disponibile anche una stazione di lavoro, con spazio per Spectrum, registratore, microdrive e monitor. Il prezzo di quest'unità è di 22.95 sterline.

Il Computer Desk della PH' Scientific Products è un tavolo in ABS adatto a Spectrum e ZX81.

Provvisto di incavi per computer e stampante, fornisce anche supporto al televisore. L'alimentatore e tutti i cavi possono essere elegantemente nascosti.

Il prezzo della versione per Spectrum è di 16 sterline, mentre quella per ZX81 ne costa una in meno.

Il Backpack per ZX81/Spectrum è disponibile in sei modelli, tutti provvisti di interruttori SAVE/LOAD e ON/OFF, nonché di spazio per l'alimentatore. Il modello più prestigioso possiede anche un amplificatore regolabile per il suono, tre prese, interruttore e spia. Da 27.50 a 12.85 sterline dalla Kelwood Computer Cases.

TASTIERE

Sono molti i modelli di tastiera aggiunti

va fra cui si può scegliere, alcuni giunti da poco sul mercato. La prima a produrre questo tipo di periferica è stata probabilmente la DK'Tronics. Il suo primo modello si adattava sia allo ZX81 che allo Spectrum, con la sola modifica delle scritte sui tasti. Questa tastiera è stata molto criticata, ma si trattava dopotutto di un buon prodotto. La nuova versione sembra abbia tenuto conto di tutti questi commenti; è infatti microdrive compatibile e possiede una barra spaziatrice di dimensioni normali. È venduta a 15 sterline.

Quest'unità possiede anche un tastierino numerico alla destra della normale tastiera.

La Transform Ltd. produce un modello apparentemente simile per 69.75 sterline, sebbene la sua costruzione appaia più accurata ed il contenitore sia in metallo.

Della Ricoll è un'ottima tastiera ad un





prezzo davvero interessante. Provvista di un contenitore metallico e di barra spaziatrice di dimensioni normali, a dispetto delle sue dimensioni compatte, costa 37.95 sterline.

All'ultima Microfire è stata presentata un'interessante tastiera Low Profile della Advanced Memory Systems.

La tastiera Saga 1 Emperor è di aspetto molto elegante e compatto. A 54.95 sterline è garantita compatibile con tutte le periferiche, inclusa l'Interface 1. Maggiori informazioni vanno richieste alla Saga, Woodham Road, Woking, Surrey.

In breve

La Tactile Keyboard della Steatite Insulations, per ZX81, possiede tasti professionali, che consentono una vera digitazione ad alta velocità.

Il contenitore ha un profilo estremamente basso ed i tasti sono inclinati, in modo da rendere più agevole la digitazione. Le scritte sui tasti sono ad intarsio, così da rimanere perfette per un tempo lunghissimo.

Sono fornite istruzioni complete per il montaggio. Si può ottenere per 30.95 sterline dalla Steatite.

Costruita su di un contenitore in ABS nero, la Push Button Keyboard si colloca sulla tastiera dello ZX81. Una volta inserita la ricopre completamente; i suoi tasti rilevati danno un'ottima sensazione. Il suo prezzo è di 9.95 sterline; maggiori informazioni su questa economica tastiera dalla Filesixty Ltd.

I Custom Key Panel Kits della Softeach permettono di modificare la vostra tastiera per adattarla ai diversi programmi usati.

In alcuni giochi, ad esempio, i tasti da premere sono l'"8" ed il "5"; altri giochi possono invece avere esigenze completamente diverse e richiedere la pressione di un numero sconcertante di tasti. Con questo kit diventa possibile sistemare una mascherina sulla tastiera ed usando etichette adesive creare chiare indicazioni per diversi programmi.

La Softeach fornisce dieci mascherine e due fogli di etichette, di cui uno già stampato con diciture come "sinistra", "destra", "laser", "fuoco", ecc., e l'altro bianco per poter creare le vostre etichette personali.

I Custom Key Panel Kits costano 3.95 sterline l'uno. Questo vuol dire però che dovrete avere due unità separate. I tasti, di tipo professionale e inclinati, non funzionano molto bene, ma la linea del contenitore è veramente elegante, nello stile Memotech.

La Computer Keyboard per ZX81/Spectrum è di aspetto un po' massiccio, ma la tastiera è davvero ottima. Anche qui occorre smontare il computer e sistemarlo all'interno, dove c'è spazio anche per l'alimentatore. È disponibile in due versioni, con contenitore standard o metallico. Standard 35 sterline, metallico 50 sterline. È prodotta dalla Keyboard.

Per coloro che hanno problemi fisici esiste una serie speciale di tastiere, comprendente il Desk-top Scanning Device, il Briefcase Scanning e l'Expanded Keyboard. Ognuno di essi è stato progettato per aiutare persone con un particolare problema. Il loro prezzo dipende molto dai congegni speciali che vanno collegati; per questo la cosa migliore da fare è chiedere informazioni alla Possom Controls Ltd., Middlegreen Road, Langley, Berks.SL3 6 DF, telefono 0753 79235.

La Microtext produce una tastiera per lo Spectrum al prezzo di 53.45 sterline. I tasti portano scritte molto chiare, alcune in cinque colori, che indicano le varie funzioni. È inoltre compresa una seconda serie di tasti cursore disposti a croce, con al centro un pulsante di fuoco.

Un nuovo concetto per quanto riguarda le maschere per tastiere è stato introdotto dalla Tactile, con particolare riguardo ai programmatori più giovani. Il Tactile Keyframe si adatta al di sopra dello Spectrum e serve da supporto ad una serie di maschere, fornite in ogni forma e colore. La stessa ditta produce software che fa uso di queste maschere.

Presentata dalla spagnola Indescomp una nuova serie, tutta in contenitori neri lucidi, molto eleganti. Comprende un nuovo Domestic Controller per lo Spectrum con quattro uscite per controllare allarmi, luci, riscaldamento e così via. Le tastiere sono tutte di tipo professionale.

La tastiera per lo Spectrum costa 42.95 sterline, per lo ZX81 39.95 sterline, il Domestic Controller 37.75.

Indirizzi

A

Advanced Digital Systems, 9 Bonchurch Road, Portsmouth, Hants. S05

8 LA

C

Cobra Technology, 378 Caledonian Road, London N1 LDY
Computer Keyboards, Glendale Park, Firmsbank Road, Ascot, Berks
Computer Cases, 4A Waters Edge, 148 Wellesley Road, Chiswick, London

D

DK'Tronics, 23 Sussex Road, Gorleston, Great Yarmouth, Norfolk

F

Filesixty, 25 Chippenham Mews, London W9 2AN

I

Indescomp P. Castellana, 179, I. Madrid 16

K

Kelwood Computer Cases, Downs Row, Moorgate, Rotherham, South Yorks. S60 2 HD

M

Memotech, Station Lane, Witney, Oxon OX8 6 BX

Microtext UK, Highland House, 18-24 John Street, Luton, Beds. LU1 2JE
Morex Peripherals, 2 Balliol Road, Caversham, Reading, Berks.

Micro Aids, Distribution Centre, Beech House, Hob Hey Lane, Culcheth, Warrington, Cheshire WA3 4 LN
Memotech, Station Lane, Witney, Oxon OX8 6 DX

P

Print'n'Plotter Products, 19 Borough High Street, London SE1 9 SE

Possum Controls, Middlegreen Road, Langley, Slough, Berks SL3 6 DF
PH Scientific Products, 9 Southfield, Welwyn Garden City, Herts.

S

Steatite Insulations, Hagley House, Hagley Road, Birmingham B16 8QW

Softeach, 25 College Road, Rading, Berks. RG6 1QE
Sinclair Research, Stanhope Road, Camberley, Surrey GU15 3 PS

T

Tactile, Wraith, 32 Emfield, Kingswood, Bristol BS15 2SS

Traffic Technology, PO Box 2, Warminster, Wilts. BA12 7 QX
Treetop Designs, 61 Widmore Road, Bromley, Kent
Technology Research, 57 Brockley Rise, London SE23

...then

Dal BASIC al Prolog: la Babele elettronica

di **Marcello Spero**

I calcolatori non parlano solo BASIC, ma molti di essi sono in realtà autentici poliglotti. Vediamo insieme, seppure sinteticamente, quali sono le altre lingue che tanti computer (fra i quali, in ottima posizione, figura lo Spectrum) possono parlare al posto del BASIC.

Quando si parla di computer i termini "linguaggio ad alto livello", "linguaggio evoluto" e simili si sprecano. Dalla nostra esperienza diretta con lo Spectrum noi sappiamo che esiste il BASIC, che fa parte dei cosiddetti linguaggi ad alto livello e che si esprime con parole, anche se, purtroppo, inglesi.

Sappiamo inoltre, alcuni per sentito dire, altri per esperienza più diretta, che il BASIC non è il solo linguaggio esistente.

I più interessati alla loro macchina, poi, sanno bene della presenza strisciante e subdola, all'interno di essa, di un altro linguaggio, ostile e tortuoso, chiamato linguaggio macchina.

Vogliamo mettere le cose in ordine?

La genesi

In principio era il codice macchina. E quando diciamo principio intendiamo davvero qualcosa di remoto; l'alba degli elaboratori elettronici risale infatti al periodo bellico, durante il quale qualcuno aveva compreso come un veloce mezzo di calcolo potesse costituire un enorme vantaggio sull'avversario in molte occasioni.

Dicevamo che allora, ed in seguito per diversi anni, esisteva solo il codice macchina. Si tratta del linguaggio fondamentale dell'elaboratore, cioè del linguaggio per comprendere il quale la

macchina è stata costruita. Composto da lunghe serie di codici numerici, che rappresentano le varie operazioni elementari che il computer può compiere, costringeva i programmatori a lunghissimi lavori di preparazione a tavolino. Occorreva infatti passare dal ragionamento di tipo umano all'impostazione logica richiesta dalla macchina.

Per questa ragione i programmatori erano pochissimi, in gran parte legati all'ambiente universitario e ad un lavoro di ricerca teorica.

Certo, era possibile esprimere il codice macchina per mezzo del linguaggio Assembly, che a ciascun codice di operazione sostituisce una sigla più comprensibile a noi poveri esseri umani - a quel tempo in condizione di profonda frustrazione -, ma questo non migliorava di molto le cose. La conversione dell'Assembly ai codici numerici andava infatti eseguita a mano, visto che l'unico modo di introdurre dati e programmi nel computer era attraverso interruttori e commutatori, non esistendo ancora tastiere come noi le conosciamo. In ogni caso, comunque, la memoria allora a disposizione non sarebbe stata sufficiente a contenere un listato Assembly, il programma traduttore ed i codici tradotti.

Con il passare degli anni qualcosa lentamente cambiò. I transistor sostituirono le valvole all'interno dei circuiti elettronici, e questo permise un aumento della quantità di memoria, prima fortemente limitata dal suo ingombro e dal consumo di energia elettrica. Un certo Holleirth inventò poi quello che sarebbe stato il cardine del progresso informatico: la scheda perforata.

Con questo nuovo mezzo non era più necessario introdurre i codici per mezzo di interruttori e commutatori, ma bastava perforarli su schede che venivano lette da un congegno apposito. Queste

schede, inoltre, erano relativamente semplici da conservare, e potevano essere rilette, e quindi riutilizzate, ogni volta che era necessario usare il medesimo programma.

In questa nuova situazione (agli albori degli anni '50) iniziò a farsi strada l'idea che fosse possibile realizzare programmi in grado di produrre codice macchina a partire da un linguaggio più adatto al modo di ragionare degli esseri umani. Cambiando semplicemente questo traduttore, inoltre, sarebbe divenuto possibile trasferire i programmi da un elaboratore all'altro, ora che le schede perforate ne consentivano il trasporto materiale. La diversità dei vari linguaggi macchina sarebbe stata così superata. Nacquero così i primi linguaggi ad alto livello.

Già nel '45 si hanno i primi esempi di linguaggi di questo tipo, ma è solo fra il '54 e il '57 che alla IBM nasce il primo vero prodotto utilizzabile, destinato a diventare non solo la pietra miliare della programmazione, ma anche il punto di partenza per la creazione di successivi, e più moderni, linguaggi. Si tratta del Fortran, contrazione delle parole Formula Translation. Lo scopo per cui era stato pensato era infatti quello di consentire l'introduzione nel computer di calcoli e formule usando la normale notazione matematica. Si tratta senz'altro di qualcosa che oggi diamo per scontato, ma chi ha provato a tradurre un calcolo anche semplice in codice macchina si sarà reso conto che la distanza fra il nostro modo di pensare ad un calcolo e quello della macchina è davvero grande.

Dicevamo del Fortran; alla sua comparsa ci furono molte critiche, poiché fino ad allora nessun compilatore (così si chiama un programma che traduce in codice macchina un linguaggio ad alto livello) era mai stato in grado di produrre

re un codice qualitativamente pari a quello dei programmi codificati a mano. Il raggiungimento di questo obiettivo, ottenuto dal gruppo della IBM, fu un risultato straordinario per quei tempi.

L'evoluzione

Veniamo ora ai giorni nostri. Abbiamo visto come l'unico linguaggio comprensibile al calcolatore sia il suo codice macchina e come sia possibile scrivere un programma direttamente in questo modo, ma si tratta di un procedimento estremamente pesante ed esposto al pericolo di errori difficilmente identificabili.

Per alleviare questi problemi è stato creato l'Assembly, che, pur sostituendo a ciascun codice di operazione un'abbreviazione di facile comprensione, conserva un rapporto di corrispondenza fissa con il codice macchina: a quella sigla (o meglio mnemonica) corrisponde sempre uno ed un solo codice di operazione.

A volte si programma ancora in Assembly, o codice assemblatore come è più corretto dire in italiano, perché si può avere un accesso diretto a tutte le risorse del calcolatore ed i programmi prodotti sono estremamente veloci ed efficienti. Esistono moderni programmi assembler che effettuano in modo raffinato tutto il lavoro di traduzione, ma resta il fatto che programmi di questo tipo sono lunghi e le possibilità di errore enormi. E soprattutto la logica di un programma in Assembly è ben più vicina a quella della macchina che non a quella umana.

A partire dal 1960 la maggior parte del

software è stata scritta per mezzo di linguaggi di programmazione (chiamarli linguaggi ad alto livello sembra oggi improprio, in quanto nessuno si sente più di considerare "linguaggio" il codice macchina e l'Assembly), visti i molti vantaggi che presentano. Un'istruzione può corrispondere a molte operazioni in codice macchina, e quindi i programmi risultano più corti e perciò più chiari. Lavorare con concetti che richiamano il problema, invece che con quelli definiti dalla macchina, riduce inoltre le possibilità di errore. Si realizza infine, come abbiamo già visto, una "indipendenza dalla macchina", cioè la possibilità di scrivere un programma che possa essere eseguito su molti calcolatori.

Un linguaggio di programmazione è ben più di un banale metodo per dire alla macchina ciò che vogliamo da lei. Esso può trasformare completamente il carattere della macchina su cui è utilizzato.

I componenti fisici di un tipico calcolatore sono i registri, le celle di memoria, l'unità centrale e così via, e quando un programmatore scrive nel linguaggio originale del calcolatore questi sono i dispositivi che deve tener presenti. Un nuovo linguaggio produce un nuovo modello di macchina, anche se in realtà l'hardware è immutato. Il programmatore può pensare ora in termini di variabili anziché di celle di memoria, di archivi di dati anziché di canali di ingresso e uscita, di formule algebriche anziché di registri e codici operativi.

Esistono centinaia, se non addirittura migliaia, di linguaggi di programmazione e di relativi dialetti. Senza altro le lin-

gue naturali della comunicazione umana sono ancora più numerose, ma, sotto certi punti di vista, fra i linguaggi di programmazione le differenze sono più marcate.

Ogni linguaggio ha la sua grammatica e la sua sintassi, il suo modo di esprimere idee. In linea di principio qualsiasi tipo di problema potrebbe essere espresso in tutti i linguaggi esistenti, ma i vari programmi risulterebbero molto diversi. Inoltre, scrivere un programma per un dato compito sarebbe molto più facile con certi linguaggi che non con altri.

Il comportamento

È importante distinguere fra un linguaggio di programmazione e una cosiddetta implementazione del linguaggio. Il linguaggio in sé è la notazione, l'insieme delle regole che definiscono la sintassi di un programma valido.

L'implementazione di un linguaggio è invece un programma che trasforma la notazione di alto livello in successioni di codici macchina.

Esistono due tipi principali di implementazione di un linguaggio: i compilatori e gli interpreti.

Un compilatore traduce tutto il testo di un programma di alto livello in codice macchina, che potrà in seguito essere eseguito indipendentemente dal compilatore stesso. L'uso di un linguaggio compilato comporta quindi tre stadi:

- 1 - stesura del programma;
- 2 - compilazione;
- 3 - esecuzione.

Ben diverso è invece il modo di agire di

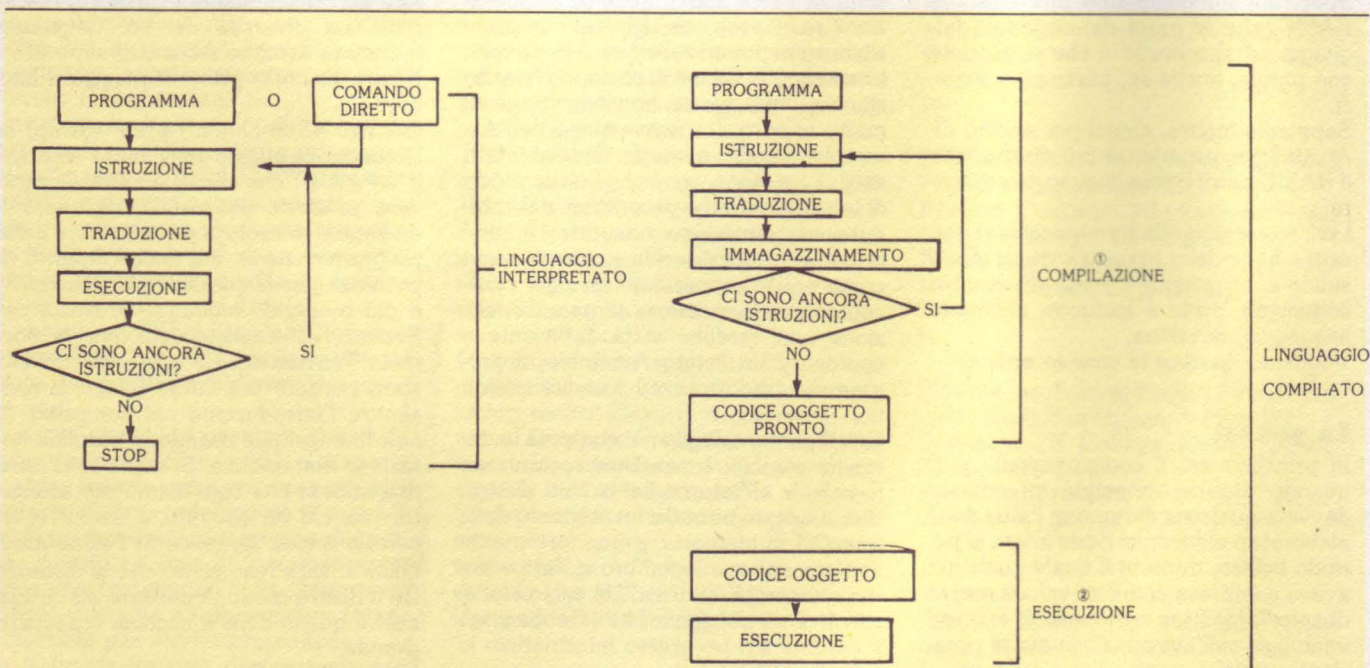


fig. 1 - Differenze fra il processo di interpretazione e quello di compilazione di un programma.

un interprete, che traduce una singola istruzione alla volta e quindi la passa all'esecuzione immediatamente. Il prodotto del suo lavoro, inoltre, non viene in alcun modo conservato, e quindi questo processo deve essere ripetuto ad ogni esecuzione del programma, o al limite ad ogni nuova esecuzione della medesima istruzione, per esempio in un ciclo.

La differenza è la stessa che intercorre fra il traduttore di un'opera letteraria e l'interprete simultaneo: il traduttore prende un testo completo e ne produce un altro, ugualmente completo, in un'altra lingua; l'interprete traduce ogni espressione o frase nel momento in cui questa viene pronunciata, per dimenticarla subito dopo.

In linea di principio qualunque linguaggio potrebbe essere interpretato o compilato, ma nella maggior parte dei casi la pratica ha fatto sì che una delle due implementazioni sia più comune dell'altra. In particolare, i piccoli calcolatori tendono ad adottare linguaggi interpretati per la minore necessità di memoria da parte di questi ultimi. Un compilatore necessita infatti dello spazio necessario a contenere entrambe le versioni del programma, e cioè quella di partenza, ad alto livello, detta sorgente, e quella di arrivo, in codice macchina, detta oggetto.

Ciascuno dei due metodi ha vantaggi e svantaggi. Un linguaggio compilato sarà sempre molto più veloce di uno interpretato, ma non consentirà un'altrettanto semplice individuazione degli errori in esecuzione né una facile correzione.

Anche l'esecuzione di un comando immediato o quella di parte del programma sono impossibili.

Le specie

Fra tutti i linguaggi, quello oggi più diffuso a livello di home e personal computer è senz'altro il BASIC. È così comune che talvolta tendiamo a dimenticarci che non è altro che uno dei tanti possibili modi per comunicare con il nostro computer.

BASIC sta per Beginner's All-purpose Symbolic Instruction Code (codice simbolico di istruzioni multiuso per principianti), ed è stato sviluppato una ventina di anni or sono da un'equipe della Dartmouth University, con il preciso intento di creare un linguaggio abbastanza vicino alla lingua inglese e quindi adatto ai principianti. Le sue parole, infatti (IF... THEN, PRINT, RUN, ecc.), sono prese direttamente da questa lingua. È normalmente implementato come linguaggio interpretato, e questo facilita enormemente la correzione ed il collaudo dei programmi.

Molto simile al BASIC, nella sua struttura, è il Fortran, di cui abbiamo già parlato, che è però un linguaggio compilato. Questo, unito al notevole ingombro dei suoi compilatori, lo rende poco adatto

ad essere utilizzato sui piccoli elaboratori. È tuttora il linguaggio preferito per le applicazioni tecniche e matematiche. La sua somiglianza con il BASIC è dovuta al fatto che i creatori di quest'ultimo lo hanno utilizzato come modello. Un altro linguaggio sempre più diffuso a livello di microcomputer è il Pascal. Battezzato con il nome del famoso matematico francese è stato creato con intenti didattici. I programmi scritti in questo linguaggio risultano infatti, grazie alla sua struttura estremamente lineare e pulita, facili da leggere e comprendere. Il programmatore che lo utilizza, per la sua stessa struttura è dissuaso dall'usare metodi poco ortodossi, tanto incoraggiati purtroppo dal BASIC, noti come "programmazione spaghettata".

Si tratta in sostanza della tendenza, nel modificare un programma, ad aggiungere GO TO piuttosto che ricorrere ad una completa riorganizzazione.

Questo crea inevitabilmente programmi contorti, che riletti dopo qualche mese saranno difficilmente comprensibili. Per questo motivo la versione originale del Pascal non possedeva l'istruzione GO TO, potendone fare a meno per mezzo di strutture molto più eleganti e chiare, che si possono riassumere nel criterio di scomposizione di un programma in tante unità minori, ciascuna con un compito preciso.

Un altro punto in cui il Pascal è intransigente sono le variabili, che non possono sorgere dal nulla come nel BASIC, ma vanno tutte dichiarate come nome e come tipo (intere, reali, booleane e così via).

Anche se può apparire un tantino "severo" e ridondante, questo linguaggio indubbiamente dà a chi lo utilizza delle ottime abitudini di programmazione. In questi ultimi tempi si sta facendo strada fra gli "home" un altro linguaggio interessante: il Forth. Chiamato così perché ritenuto dai suoi creatori il linguaggio della quarta (fourth) generazione dei computer, ma memorizzabile sull'IBM 1130, usato per scriverlo con un nome non più lungo di cinque lettere, è stato progettato originariamente per controllare il movimento di un telescopio. Oggi è largamente utilizzato in applicazioni di controllo e automazione. La sua applicazione recente più pubblicizzata è stata il controllo delle telecamere durante le riprese dei film appartenenti alla saga delle "guerre stellari" e di "Tron".

Si tratta, come il BASIC, di un linguaggio interpretato, e che può quindi eseguire non solo programmi, ma comandi

diretti. Mentre però con il BASIC siamo costretti ad usare le sole istruzioni disponibili all'origine, il Forth consente all'utilizzatore di adoperare le istruzioni esistenti per crearne di nuove, secondo le sue esigenze.

Il suo unico svantaggio, se così si può chiamare, sta nel fatto che essendo stato creato per un uso da parte di specialisti non si è badato molto ad alcune strutture particolari che possono creare qualche difficoltà. Una di queste è la RPN (Reverse Polish Notation - notazione polacca inversa) usata per le espressioni algebriche, che riunisce tutti gli operandi da una parte della linea e gli operatori dall'altra, rendendo un po' acrobatica la comprensione della loro interazione. In compenso ha come struttura centrale dei dati lo stack (catasta), che rende i programmi estremamente snelli e dinamici.

Ancora più concisa di quella del Forth è la sintassi dell'APL (A Programming Language) che nella sua versione originale richiedeva però addirittura una tastiera particolare, a causa dell'elevato numero di simboli speciali che usa. D'altra parte un linguaggio capace di sommare tutti gli elementi dispari (per esempio) di una matrice con un unico enunciato non è certo da trascurare. Recentemente ne è stata prodotta una versione per il QL, dove i simboli sono stati sostituiti da parole. La combinazione dell'APL e del QL ha senz'altro creato uno dei più potenti sistemi oggi disponibili a livello di microcomputer.

Il Lisp sotto alcuni aspetti è il più semplice dei linguaggi presi in considerazione fin qui. Ha un solo tipo di istruzione, per la precisione una chiamata di funzione. La sua grande potenza sta nel fatto che il valore prodotto da una funzione può essere un'altra funzione. Sviluppato verso la fine degli anni '50 al MIT è da allora il linguaggio preferito da coloro che si interessano di intelligenza artificiale. Il suo nome deriva da List Processing, elaborazione di liste, in quanto sia i programmi che i dati sono strutturati come liste.

Un altro linguaggio in questo campo è il Prolog, PROgramming in LOGic. Una sua versione adatta ai microcomputer, chiamata MicroProlog sta iniziando ad essere disponibile. Le possibilità in campo di AI (intelligenza artificiale) sono praticamente le stesse del Lisp.

Altri linguaggi di programmazione meritevoli di citazione sono ad esempio il C, con cui è stato scritto (in "sole" trecentomila istruzioni) il sistema operativo UNIX, molto utilizzato sui computer che vanno oltre la fascia dei personal; Ada



(dal nome della figlia di lord Byron, contessa di Lovelace e considerata simbolicamente la prima programmatrice della storia), linguaggio estesissimo commissionato dal Pentagono; infine il Modula-2, seconda creatura di Niklaus Wirth, inventore del Pascal, in effetti una versione avanzata di quest'ultimo linguaggio.

Conclusioni

Moltissimi sarebbero i linguaggi che meritano di essere ricordati. Di proposito abbiamo evitato di parlare del Logo, già ampiamente presentato in questa rubri-

ca, ma un argomento simile non può certo dirsi esaurito in queste poche pagine.

Quello che dovrebbe essere emerso è che i linguaggi di programmazione sono molti diversi fra loro ed è impossibile stabilirne una graduatoria di qualsiasi tipo. Non esiste un linguaggio di programmazione migliore di tutti gli altri, esattamente come non esiste una lingua naturale migliore delle altre. "Parlo spagnolo a Dio, italiano alle donne, francese agli uomini e tedesco al mio cavallo", diceva Carlo V (probabilmente in francese). Anche un linguaggio di programmazione va scelto in funzione dell'obiettivo che ci si prefigge.



È nato ZX Stand

L'utilità di applicazione di un nuovo accessorio-complemento allo Spectrum.

Che lo Spectrum scaldi molto non è una novità, confermata del resto anche da notizie avute di guasti conseguenti a tale calore. In conseguenza di ciò, e tenendo presente che non è consigliabile installarvi ventilatori o altri sistemi di circolazione dell'aria (in quanto l'apporto di aria esterna significa anche apporto di polvere, insidiosa nemica delle circuiterie dei computer), è stato progettato lo ZX Stand. Lo Spectrum, si sa, è da considerarsi tuttora una valida macchina perché, oltretutto, dispone già di tante periferiche ed optional vari, nonché di una consistente quantità di software. Ma il già citato calore, che si sviluppa con l'uso intenso, rischia di oscurare un po' queste sue grandi qualità: in effetti, più l'alimentatore viene sovraccaricato attraverso il collegamento di periferiche e più aumentano le possibilità che la nostra prediletta e tanto diffusa "scatoletta" vada in tilt, bloccandosi nel bel mezzo dell'esecuzione di un programma. Ciò ci costringe, quantomeno, a spegnere il computer per lasciar raffreddare il tutto e quindi riprendere le operazioni dall'inizio, ricaricando, magari dopo essersi rassegnati ad aver perso dei dati preziosi non ancora salvati. Tutti questi inconvenienti sono risolti dalla applicazione allo Spectrum dello ZX Stand, che si accolla l'onere

di dissipare completamente tutto il calore in più, con qualunque carico e qualunque temperatura ambiente, lasciando immutati i componenti originali del computer, senza che all'interno dello stesso e dei suoi delicati circuiti si depositi la polvere. In tale situazione lo Spectrum può lavorare per tempi molto lunghi, senza noie.

Osservando la foto si nota la forma particolare dello ZX Stand, che consente di utilizzare lo Spectrum in posizione ergonomica, unitamente a tutte le periferiche esistenti (compresa la ZX Interface 1) ed alle eventuali future.

Il computer stesso è infatti appoggiato sopra lo Stand e non attaccato; l'unico punto di giunzione tra i due apparecchi è costituito dal cavetto visibile nella foto, che, naturalmente a computer spento, può essere scollegato, permettendo così la separazione dei due elementi.

Lo ZX Stand è inoltre provvisto di un interruttore generale, che funge da RESET, evitando la noia di dover collegare e scollegare il cavetto di alimentazione ogni volta che l'apparecchio deve essere acceso o resettato; questo accorgimento è senz'altro apprezzabile dall'utente. Lo ZX Stand contiene una grande quantità di spazio libero, utile per ospitare, per fare degli esempi, accumulatori tampone o condensatori elettrolitici di grande capacità, per proteggersi da possibili sbalzi di tensione o interruzioni nella rete elettrica, stabilizzatori di tensione supplementari, nel caso si voglia abbassare la tensione dell'alimentatore a valori più vicini ai 9V

nominali, ecc. Come si può capire, lo ZX Stand fornisce all'utente la grande comodità di poter utilizzare il computer con estrema tranquillità, al riparo da qualunque inconveniente derivante da surriscaldamenti della circuiteria interna, a un prezzo davvero alla portata di tutti. L'abbinamento dello ZX Stand allo Spectrum permette di esaltarne al massimo le capacità, come del resto risulta dalle prove effettuate, consentendogli (e consentendovi) di operare al meglio delle sue (vostre) possibilità.

Per maggiori informazioni sullo ZX Stand rivolgetevi a:

Francesco DI MARCO
Via F.lli Rosselli, 9
33051 Aquileia (UD)
Tel. 0431/91461
Telex 450416 MAHL I

Visione di insieme dello Spectrum connesso allo ZX Stand



REM:HW

Hardware

di **Marcello Spero**
Prima parte

Estendiamo le possibilità della nostra interfaccia con un convertitore analogico/digitale

Ci avviciniamo alla fine della nostra serie sulla costruzione di un'interfaccia per lo Spectrum.

Eccoci finalmente al "tocco finale", ovvero all'ultimo pezzo della nostra ormai quasi mostruosa costruzione. Prima che, con un gesto di definitiva ribellione, voltiate pagina con la ferma intenzione di mai più tornare a leggere questa rubrica, vediamo di capire insieme cos'è e a cosa può servire un convertitore A/D.

Finora il nostro Spectrum è stato in grado di ricevere, per mezzo dell'interfaccia da noi realizzata, varie informazioni provenienti dal mondo esterno. La posizione di un joystick, l'apertura o chiusura di un interruttore, la condizione di illuminazione o meno di una fotocellula sono solo alcune delle possibili "conoscenze" che il computer era in grado di acquisire.

Se però consideriamo attentamente la natura di queste informazioni, ci accorgiamo di come esse siano di un unico tipo, e cioè DIGITALI. La parola "digitale" (che non deriva da dito, ma da "digit": "cifra" in inglese) sta ad indicare una quantità che non può variare in modo continuo, ma solo a "scatti" tutti uguali, che ne rappresentano l'unità di misura.

Il numero dei possibili "scatti" dipende dal numero di cifre binarie, in inglese Binary digit abbreviato in bit, da cui l'informazione è composta, secondo la formula

$$N=2^B$$

dove N è il numero di "scatti", cioè di

valori diversi che possono essere assunti, e B è il numero di bit che compongono l'informazione.

Un interruttore che si apre e si chiude produce un'informazione digitale ad un solo bit e quindi capace di assumere due soli valori, che possiamo chiamare "uno" o "zero", "sì" o "no", o come preferite. Questi due soli valori, comunque, bastano a rappresentare i possibili stati di un interruttore, che sono solo due: "aperto" e "chiuso" (o "acceso" e "spento"). Lo stesso discorso vale per una fotocellula, che può essere solo illuminata o no.

Il caso del joystick è più interessante. Qui, infatti, vengono utilizzati 5 bit per trasmettere l'informazione relativa alla sua posizione. Applicando la formula, otteniamo

$$2^5=32$$

possibili valori diversi. Come vedete dalla figura 1, però, le diverse combinazioni posizione-fuoco che un joystick può assumere sono solo 16. Ciò vuol dire che, per privilegiare la semplicità costruttiva, è stata adottata una codifica "ridondante" dell'informazione, che di per sé avrebbe richiesto solo 4 bit. Invece le informazioni contenute in memoria, come è noto, possiedono una gamma di 256 possibili valori, corrispondenti ai numeri 0-255. Per questo sono codificate con 8 bit:

$$2^8=256$$

Tutte le informazioni che circolano all'interno di un computer sono di tipo digitale, e dello stesso tipo devono necessariamente essere anche quelle che

gli giungono dall'esterno. Tutto bene, quindi, per interruttori, joystick e simili, ma le informazioni disponibili all'esterno sono anche di un altro tipo.

In natura, infatti, non esistono (tranne che a livello subatomico, ma la cosa non ci interessa molto, in questo momento) grandezze che variano "a scatti", in modo digitale. Il tempo, le stagioni, la temperatura variano tutti in modo continuo, cioè ANALOGICO.

Pare però che a noi esseri umani le grandezze di tipo analogico non vadano troppo a genio. Infatti ci sforziamo di esprimerle in modo digitale, non potendo ottenere altro che una loro approssimazione più o meno precisa.

Domanda:

"Che ore sono?"

Possibili risposte:

1 - "Circa le quattro."

- 1 nord
- 2 nord e fuoco
- 3 nord est
- 4 nord est e fuoco
- 5 est
- 6 est e fuoco
- 7 sud est
- 8 sud est e fuoco
- 9 sud
- 10 sud e fuoco
- 11 sud ovest
- 12 sud ovest e fuoco
- 13 ovest
- 14 ovest e fuoco
- 15 nord ovest
- 16 nord ovest e fuoco

fig.1 - le varie possibili combinazioni delle posizioni di un joystick.

Approssimazione grossolana, che possiamo definire a 5 bit. Infatti i possibili valori di questa approssimazione sono solo 24 (le ore del giorno), ed il numero minimo di bit necessario a rappresentarli tutti è 5:

$2^4=16$ insufficiente
 $2^5=32$ abbondante

2 - "Le quattro e dieci."

Approssimazione al minuto, sufficiente per la nostra vita normale. Può assumere 60×24 valori diversi, per un totale di 1440. Si tratta di un'approssimazione a 11 bit:

$2^{10}=1024$ insufficiente
 $2^{11}=2048$ abbondante

3 - "Le quattro, dieci minuti e diciotto secondi."

Approssimazione "cronometrica", con gamma di possibili valori pari a $24 \times 60 \times 60 = 86400$. I bit necessari sono 17:

$2^{16}=65536$ insufficiente
 $2^{17}=131072$ abbondante

Si potrebbe andare avanti all'infinito, con un aumento lineare del numero di bit, esponenziale della possibile gamma di valori ed un avvicinamento sempre maggiore alla realtà, che però potrebbe essere raggiunta solo con una rappresentazione ad infiniti bit, e quindi mai. Lo stesso discorso vale per la temperatura, che siamo abituati ad esprimere in gradi, decimi, centesimi (???), avvicinandoci sempre più, con una scalinata composta da gradini sempre più piccoli, a ciò che invece è una salita liscia ed uniforme.

Dopo questa allucinante dissertazione, che ci porta filosoficamente alla desolante conclusione che la perfetta conoscenza della realtà ci sarà sempre preclusa, potendo disporre soltanto di una sua approssimazione sempre più precisa, torniamo al nostro Spectrum.

Ciò che ci proponiamo di fare è realizzare un congegno che consenta al computer di conoscere grandezze che in origine sono di tipo analogico, fornendogliene approssimazioni digitali più o meno precise, a seconda del livello di perfezione del congegno stesso. In pratica, all'ingresso di questo congegno ci sarà qualcosa che varia in modo continuo, mentre all'uscita troveremo un numero che ne rappresenta un'approssimazione del valore. Quante più cifre avrà questo numero, tanto migliore sarà la rappresentazione di ciò che è all'ingresso.

Il metodo scelto è quello che ci permette di realizzare il tutto nel modo più semplice possibile, e si basa su di un concetto molto intuitivo.

Supponiamo di disporre di un rubinetto e di un secchio. Mettiamo il secchio sotto il rubinetto ed apriamo quest'ultimo:

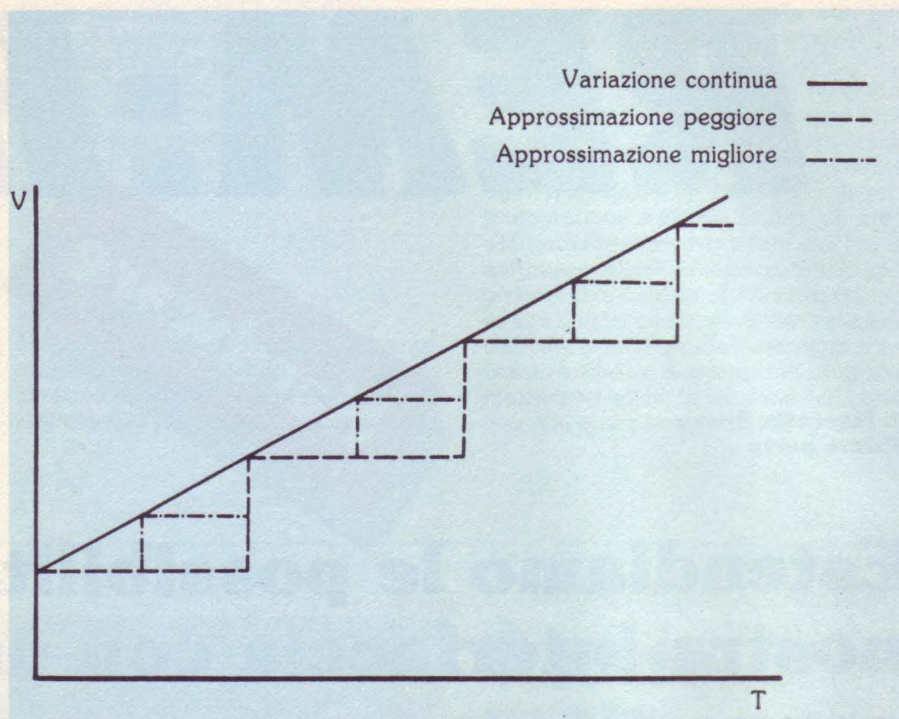


fig.2 - esempio grafico dell'approssimazione digitale di una grandezza analogica.

l'acqua inizierà a scorrere ed il secchio si riempirà, più o meno velocemente. Il tempo necessario affinché il secchio si riempia completamente dipenderà ovviamente da quanto abbiamo aperto il rubinetto.

Se a questo punto facciamo delle tacche sul rubinetto, e cronometriamo il tempo occorrente a raggiungere le varie tacche, otterremo una tabella come quella che vedete in figura 3.

Disponendo di questa tabella, potremo sempre sapere, in base al tempo impiegato dal secchio a riempirsi, in che posizione era il rubinetto, con approssimazione pari ad una tacca.

Avvicinando sempre di più le tacche fra loro, raggiungeremo un punto in cui il nostro cronometro non sarà più abbastanza preciso, le variazioni anche piccole nella pressione dell'acqua introdurranno degli errori e così via.

Saremo giunti cioè ai limiti di precisione del sistema. Il numero di tacche che saremo riusciti a segnare fino a questo momento rappresenterà la massima gamma di valori che possono essere distinti l'uno dall'altro, e quindi il numero di bit dell'approssimazione.

Sostituendo all'acqua la corrente elettrica, al rubinetto una resistenza variabile ed al secchio un condensatore ci saremo fatti un'idea molto precisa sul funzionamento del convertitore A/D (Analogico/Digitale) che vogliamo costruire.

In pratica, abbiamo una corrente elettrica che si riversa, attraverso una resistenza, in un condensatore. Questo, però, per il momento non si "riempie";

è come se fosse "aperto il tappo". Nel momento in cui facciamo partire il cronometro chiudiamo il "tappo" ed il condensatore inizia a riempirsi. Nel preciso istante in cui questo risulta "pieno" per $2/3$ della sua capacità scatterà un segnale, che bloccherà il cronometro e toglierà il "tappo", provocando un rapido svuotamento del condensatore. Dal tempo impiegato per il riempimento potremo dedurre il grado di apertura del rubinetto, cioè il valore della resistenza.

Considerato che esistono resistenze variabili con la temperatura, con l'umidità, con la pressione, con la luce e col variare di tantissime altre grandezze ambientali, un simile sistema è piuttosto versatile.

Il circuito vero e proprio, di cui vedete lo schema in figura 4, è molto semplice. Tutto il lavoro è infatti svolto dal piccolo circuito integrato 555.

RUBINETTO APERTO ALLA TACCA N.	TEMPO DI RIEMPIMENTO
1	15 min.
2	12 min.
3	9 min.
4	6 min.
5	3 min.
6	1 min.

fig.3 - tabella dei vari tempi di riempimento in corrispondenza di varie tacche praticate su di un rubinetto.

Si tratta di un integrato dalle caratteristiche davvero interessanti e dagli usi più disparati. In figura 5 ne vedete lo schema a blocchi. Vediamo rapidamente quali sono gli elementi che lo compongono e le loro funzioni.

COMPARATORE INFERIORE (piedino 2)

Quando è portato a livello basso (meno di 1/3 della tensione di alimentazione, nel nostro caso +5V) "chiude il tappo", cioè toglie il collegamento a massa del terminale superiore del condensatore; inoltre porta a livello alto (+5V) l'uscita.

COMPARATORE SUPERIORE (piedino 6)

Quando è portato a livello alto (più di 2/3 della tensione di alimentazione) porta l'uscita a livello basso (circa 0V) e "toglie il tappo", cioè collega a massa il terminale superiore del condensatore, provocandone la scarica.

FLIP-FLOP

Fa passare l'uscita dal livello alto a quello basso, e viceversa, ed inoltre comanda il transistor che "tappa" e "stappa" il condensatore.

STADIO DI USCITA (piedino 3)

Segnala con il suo livello la fase di carica o di scarica del condensatore.

TENSIONE DI CONTROLLO (piedino 5)

Serve a variare il punto di "pieno" del condensatore rispetto ai normali 2/3 della tensione di alimentazione.

RESET (piedino 4)

Se è portato a livello basso, interrompe l'eventuale fase di carica del condensatore, facendolo scaricare e portando l'uscita a livello basso.

La figura 6 schematizza invece il modo in cui viene utilizzato il 555 nell'uso che ne facciamo noi, mentre la figura 7 riporta i segnali presenti ai piedini che ci interessano durante i vari stadi del ciclo di funzionamento.

Come vedete i componenti esterni sono solo tre: il condensatore C1, la resistenza incognita R1 (cioè quella di cui si vuole conoscere il valore) ed un'altra resistenza R2, per garantire che l'ingresso del comparatore inferiore sia a livello alto quando non lo "forziamo" noi a livello basso. Il terminale inferiore del condensatore è collegato a massa, mentre quello superiore si connette sia al comparatore superiore, che deve rilevare la condizione di "pieno", che al piedino di scarica, che si occupa dello "svuotamento".

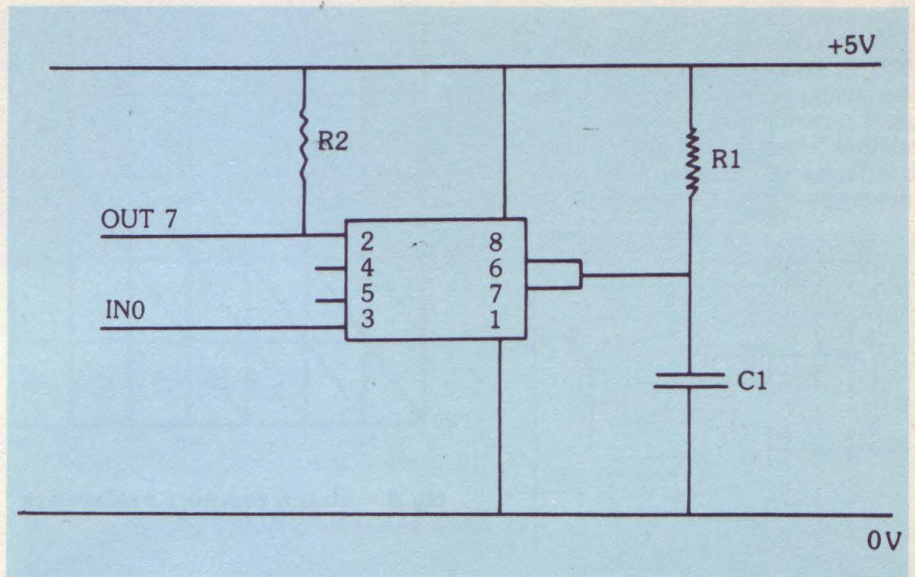


fig.4 - schema del circuito.

Ovviamente la resistenza incognita (il "rubinetto") è collegata fra la tensione positiva (il "tubo dell'acqua") ed il condensatore (il "secchio"). I piedini di reset e della tensione di controllo non vengono utilizzati.

La capacità del condensatore (si chiama proprio capacità, come quella di un recipiente e si misura in Farad, o meglio con i suoi sottomultipli) va scelta ovviamente a seconda dell'ordine di grandezza della resistenza incognita e del tempo che si desidera venga impiegato per ogni carica. La figura 8 è appunto un abaco, che mette in relazione queste tre grandezze.

La scelta fatta da noi, 0.47 microfarad, dovrebbe rappresentare un buon compromesso, ma nulla vi vieta di variarla a piacimento.

Per farvi un'idea della precisione del nostro "cronometro", che condiziona il tempo minimo di carica possibile, tenete conto che useremo a questo scopo una routine in linguaggio macchina e che ciascuna unità di conteggio durerà circa 50 cicli. Alla frequenza di clock dello Spectrum, 3.5 MHz, questo significa

$$(1/3.5E6) \cdot 50 = \text{circa } 15 \text{ microsec.}$$

In pratica, con il valore scelto da noi si ottengono misurazioni soddisfacenti a partire da un migliaio di ohm fino a circa 100 Kohm.

E veniamo finalmente alla costruzione pratica. Il piano di montaggio lo trovate in figura 9, con i valori per i due (!!!) componenti. A questo proposito un'osservazione che riguarda il condensatore. Alla stabilità di questo componente è affidata la precisione di tutto il siste-

ma. Quello che vedete montato sul prototipo è del tipo a poliestere e non ha dato risultati troppo buoni. Il consiglio è di usare senz'altro un condensatore al tantalio. Poco importa se il valore non è esattamente quello che volete voi: l'importante è che resti costante nel tempo e con la temperatura. **NON USATE CONDENSATORI ELETTRONICI!**

Come vedete, è stata usata una delle solite basette a bollini, con una presa adatta a collegarsi con lo stadio di ingresso dell'interfaccia. Di qui passerà il segnale di fine conteggio, quello cioè che arresterà il "cronometro" a conden-

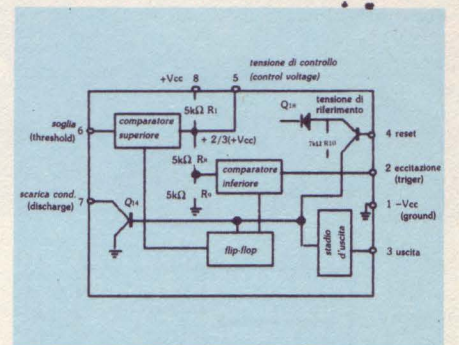


fig.5 - schema a blocchi del circuito integrato 555.

satore pieno. A questo scopo viene utilizzato IN 0.

Il segnale di inizio carica verrà invece inviato dal computer, in concomitanza con l'inizio del conteggio. Per questo occorre collegarsi alla sezione di uscita. Questo è stato fatto non con un'altra grande presa, ma con un piccolo spinotto ad un contatto, che va a collegarsi

ad un corrispondente spinotto posto sulla basetta dei relay, in collegamento con OUT 7, finora inutilizzato. Per quanto riguarda il circuito integrato quasi certamente lo troverete in un contenitore "dual in line" a 8 piedini, che è la forma rappresentata nel piano di montaggio e la prima di quelle illustrate

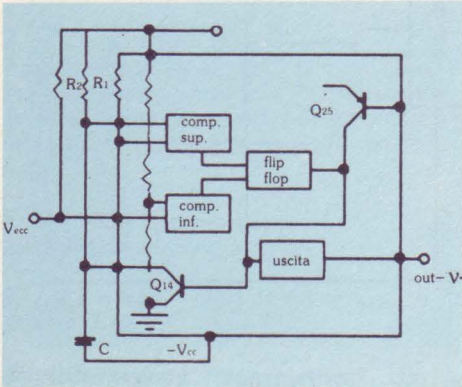


fig. 6 - schema a blocchi del 555 nel mondo di funzionamento utilizzato nel circuito.

in figura 10. Le altre due forme, benché più rare, vi possono capitare: fate attenzione a far corrispondere i piedini con lo stesso numero nel montaggio. Per la saldatura del circuito i casi sono due: o avete acquisito precisione e rapidità nel compiere questa operazione o vi conviene usare il solito zoccolo. A questo proposito è bene tener presente che un 555 "scottato" può essere ancora funzionante, ma solo a frequenze di carica e scarica molto basse. Come vedete dalle foto e dal piano di montaggio, i due punti cui va collegata la resistenza incognita sono stati collegati con una morsettiera a due contatti. Questo facilita molto il collegamento dei vari tipi di sensore che occorreranno di volta in volta.

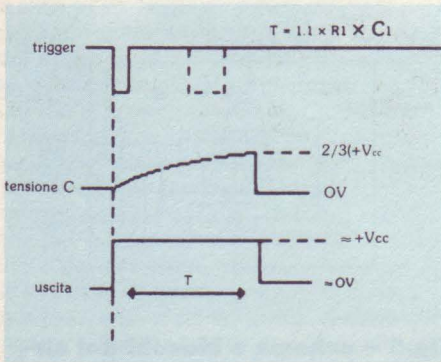


fig. 7 - ciclo di funzionamento del 555.

L'alimentazione, se avete utilizzato un connettore del solito tipo, arriva direttamente dall'interfaccia e non presenta quindi problemi. Una volta terminato il montaggio, e ricontrollati tutti i collegamenti, possiamo

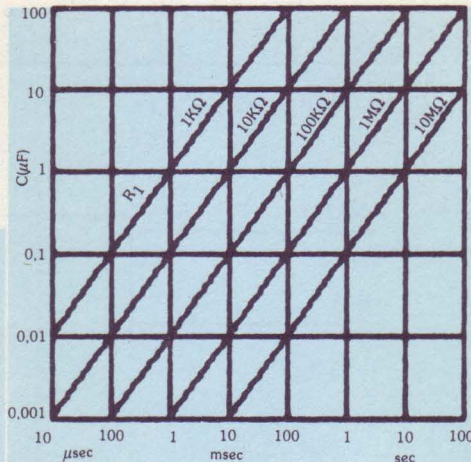


fig. 8 - abaco tempo / resistenza / capacità per il 555.

passare ad una prova pratica. Per il momento sarà solo una prova volta ad accertare il corretto funzionamento dell'insieme, e quindi ricorremo ad un'approssimativa routine di temporizzazione in BASIC.

Inserite il convertitore nel connettore di ingresso, quindi collegate lo spinottino alla presa che avrete fissato sulla piastrina dei relay e collegato ad OUT 7. Collegate anche la piastrina dei relay al suo connettore.

Inserite ora l'interfaccia allo Spectrum, e quindi procedete all'accensione nel solito ordine.

Procuratevi ora un potenziometro, cioè la "manopola del volume" di qualche vecchio apparecchio, od un paio di resistenze di valore molto diverso, ma compreso fra 1000 e 100.000 ohm. Nel primo caso collegate ai morsetti il polo centrale ed uno dei due laterali del po-

tenziometro. Nel secondo caso collegate invece i due capi di una delle due resistenze. Fate girare adesso questo programma:

```

10 OUT 223,128
20 LET i=0
30 OUT 223,0
40 OUT 223,128
50 IF IN 223 < > 0 THEN LET i=i+1:
GO TO 50
60 PRINT AT 10,10;i;" "
70 GO TO 20

```

Sullo schermo comparirà un numero, che dovrebbe essere stabile. Se ora ruotate il comando del potenziometro, o sostituite la resistenza collegata con un'altra, questo numero dovrebbe cambiare: verso valori minori per resistenze più basse, o posizioni del potenziometro più spostate nella direzione del polo laterale collegato, verso valori maggiori in caso contrario. Se non collegherete nulla ai morsetti, il programma girerà all'infinito senza visualizzare alcun numero.

La cosa importante è che appaiano dei numeri e che varino al variare della resistenza applicata. Se questo si verifica, tutto è a posto, e la prossima volta procederemo ad un controllo più accurato in linguaggio macchina, nonché ad una taratura.

In caso abbiate problemi, verificate prima di tutto che il collegamento fra morsetti e resistenza o potenziometro sia buono, e quindi rivedete il montaggio. Tenete presente che il 555 non deve assolutamente scaldare.

La prossima volta vedremo qualche applicazione che non mancherà di stupirvi. Arrivederci!

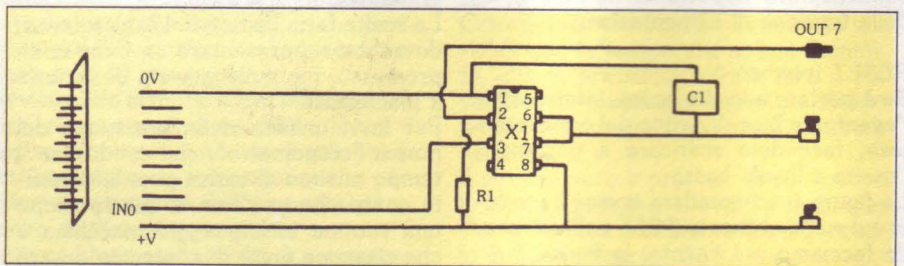


fig. 9 - piano di montaggio del convertitore.

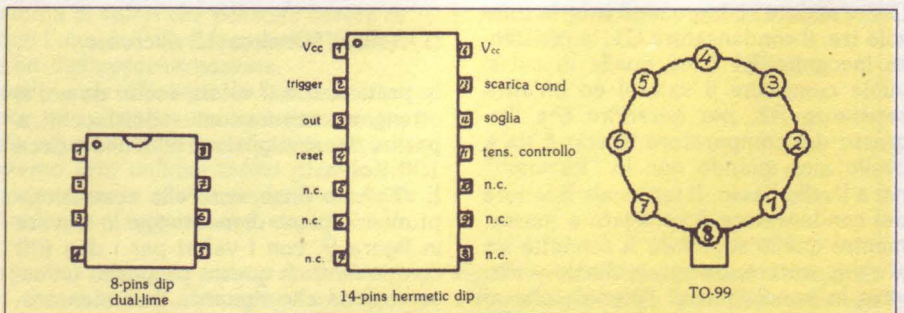


fig. 10 - i vari contenitori in cui è prodotto il 555.

RANDOMIZE

L'Assembly assieme

di **Marcello Spero**

Seconda parte del nostro piccolo corso di linguaggio macchina per utenti di Spectrum e ZX81.

Eccoci di nuovo ad esplorare i misteri del linguaggio macchina. La scorsa volta abbiamo visto come sia possibile caricare un dato in un registro a 8 bit o in una coppia di registri vista come un unico grande registro a 16 bit. Nel primo caso il massimo valore possibile è 255, nel secondo 65535. Consigliando a quanti avessero dubbi riguardanti questi argomenti di dare una ripassatina alla puntata precedente, proseguiamo.

Vi ricordate?

Quelli di voi che hanno provato a far eseguire la routine che concludeva la puntata avranno notato come il valore che compare sullo schermo al momento del ritorno al BASIC sia maggiore di uno rispetto al dato caricato nel registro bc, o hl per lo ZX81 (a proposito: potremmo benissimo usare, in questi esempi, bc anche per lo ZX81; poiché però questo computer stampa, al termine dell'esecuzione del l/m, non il valore di bc, ma quello di hl, non ci servirebbe a molto!). Questo è l'effetto del nuovo codice operativo usato:

inc

abbreviazione di "Increment", incrementare. Esso infatti aumenta di 1 il valore del registro, o della coppia di registri, che lo segue. Così

ld a,1
inc a

produrrà nel registro a il valore 2.

ld bc,1
inc bc

darà il medesimo effetto. Lo stesso discorso vale, al contrario, per

dec



che provoca la diminuzione di 1 del valore del registro, o coppia di registri, indicato subito dopo. Quindi,

ld a,1
dec a

darà in a il valore 0, e così pure

ld bc,1
dec bc

Qualcosa di nuovo

Proviamo tutto quanto, più qualcosa di nuovo, in questa routine:

ld a,0	62	0
inc a	60	
ld c,a	79	
dec a	61	
ld b,a	71	
ret	201	

o, per lo ZX81:

ld, a,0	62	0
inc a	60	
ld l,a	111	
dec a	61	
ld h,a	103	
ret	201	

Cosa succede?

Sullo schermo leggeremo il numero 1; perché?

Seguiamo lo svolgimento delle varie operazioni, una per una.

1- ld a,0

Al registro a viene dato il valore 0.

2- inc a

Il registro a viene incrementato; si ha cioè l'equivalente dell'istruzione BASIC

LET a=a+1

3- ld c,a

Il contenuto di a viene copiato nel registro c, con l'equivalente di

LET c=a

in BASIC. Si tratta di un'operazione che usiamo per la prima volta. Il contenuto di un registro qualunque può essere copiato in un qualsiasi altro registro.

4- dec a

Il registro a viene decrementato, compiendo cioè l'equivalente di

LET a=a-1

Il suo valore torna quindi a 0.

5- ld b,a

Il nuovo valore di a viene copiato in b.

6- ret

si ritorna al BASIC (tutto questo vale ovviamente anche per lo ZX81, anche se viene impiegata la coppia hl al posto di bc).

Abbiamo visto come sia possibile copiare il valore di un registro in un altro. Precisiamo meglio gli effetti dell'operazione:

ld b,a

sta per

load b with a

cioè

carica b con a

Quello che c'era prima in b viene quindi irrimediabilmente perduto, sostituito dal valore di a. Il registro a, invece, conserva intatto il suo valore, pur avendolo trasmesso a b. Quindi

ld c,h

ld a,e

ld d,l

e persino

ld a,a

ld h,h

(che non servono a niente, ma esistono per ragioni costruttive) sono tutte operazioni possibili e corrispondono ad altrettanti codici operativi.

NON ESISTONO invece, purtroppo, operazioni tipo

ld bc,de

ld hl,bc

e così via. Il trasferimento da una coppia di registri all'altra in un'unica operazione non è previsto. Per trasferire il valore di bc in hl dovremo quindi agire in due fasi:

ld h,b

ld l,c

Questo problema non esiste invece nel caso di

inc

e

dec

Sono perfettamente possibili, infatti,

inc bc

inc hl

dec de

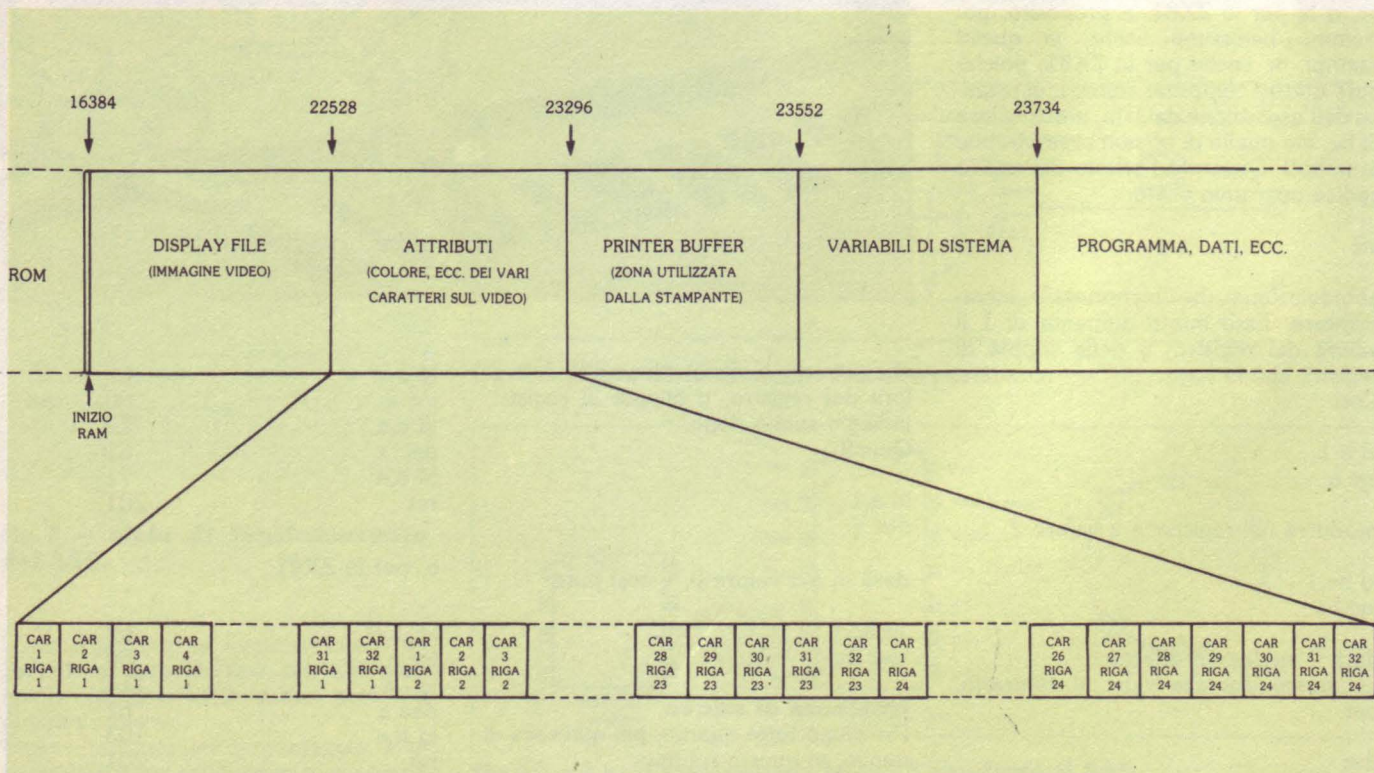
e così via. Per fortuna, visto che l'incremento o il decremento in due fasi è piuttosto complesso.

Da dove arrivano i dati?

Abbiamo visto come sia possibile eseguire operazioni, per ora elementari, sui vari registri. Ma i dati, i valori contenuti in questi registri, da dove provengono?

Semplice, dal codice seguente a quello che indica il tipo di operazione, cioè dal byte di memoria successivo.

Semplice, ma non sempre comodo. Il valore da caricare, infatti, potremmo



anche non conoscerlo, ma limitarci a sapere che si trova in un punto particolare della memoria; per esempio alla locazione 30000.

In questo caso non possiamo certo usare

ld <registro>, N

perché non sapremmo cosa mettere al posto di N. Dobbiamo usare invece un caricamento di tipo diverso:

ld <registro>, (NN)

in cui il registro verrà caricato con il valore contenuto nell'indirizzo di memoria specificato dal numero all'interno delle parentesi. Nel nostro caso

ld <registro>, (30000)

Il numero all'interno della parentesi è indicato con NN poiché è a 16 bit.

Le locazioni di memoria, infatti, sono ben più di 256 (massimo valore raggiungibile da un numero a 8 bit, contando anche lo 0). Per la precisione, anzi, la massima quantità di memoria che lo Z80 può possedere è 64K (che è la memoria totale presente nello Spectrum 48K; infatti 16K di ROM più 48 di RAM danno 64K totali). Poiché un K, o Kilo-byte, è uguale a 1024 byte, avremo

$$64 \cdot 1024 = 65536$$

che è il numero massimo di locazioni

indirizzabili con 16 bit.

Un numero a 16 bit, cioè a due byte, sarà quindi NECESSARIO e SUFFICIENTE ad indicare uno qualsiasi dei possibili indirizzi di memoria.

Tornando alla nostra istruzione di caricamento, noterete che è stato indicato un generico <registro>. Ciò non è corretto, poiché esiste in realtà un unico registro con cui è possibile effettuare questo tipo di operazione, ed è a. Come vedremo in seguito, sono molte le cose che solo questo registro, che prende il nome di ACCUMULATORE, può fare.

Per quanto riguarda il nostro esempio, quindi, l'unica forma possibile è

ld a, (30000)

Ovviamente esiste anche l'operazione inversa, cioè

ld (NN), a

che dà ad un determinato indirizzo di memoria il valore contenuto in a.

Proviamo. Carichiamo, con il solito programmino visto l'altra volta, questa routine:

```
ld a,(30000)      58 48 117
ld b,0           6 0
ld c,a           79
ret              201
```

o, per lo ZX81:

```
ld a,(30000)      58 48 117
ld h,0           38 0
ld l,a           111
ret              201
```

Poi, prima di dare

PRINT USR

stabilite il valore della locazione 30000 con

POKE 30000,54 (per esempio)

Se ora fate eseguire la routine in linguaggio macchina, vedrete apparire sullo schermo il numero 54. Perché? Quello che abbiamo fatto è l'equivalente di una istruzione

PRINT PEEK 30000

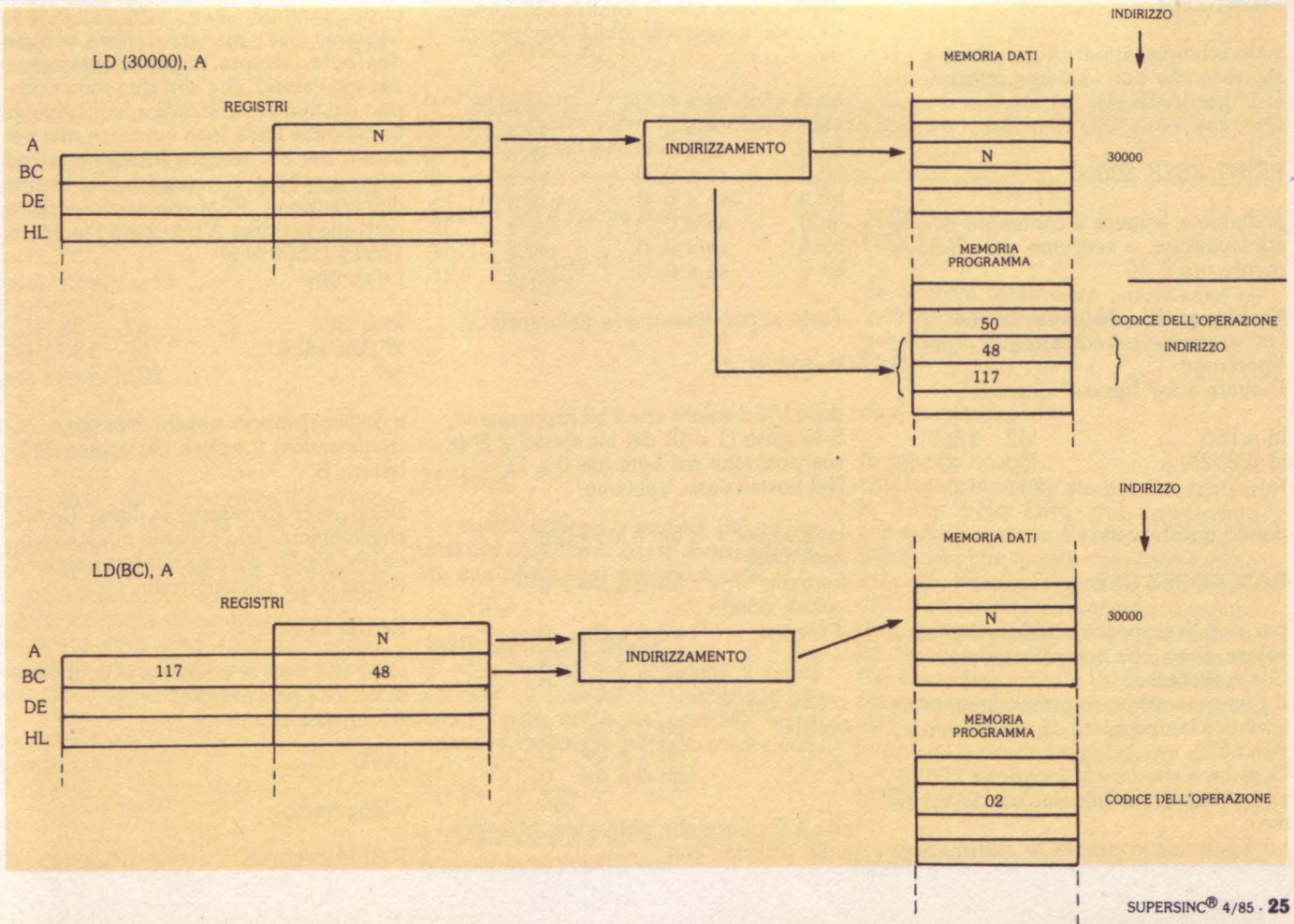
in BASIC. Infatti, la prima operazione, cioè

ld a, (30000)

trasferisce in a il contenuto di questa locazione. Notate il solito laborioso metodo per indicare con due codici il valore di un numero a due byte, in questo caso 30000. Nel primo codice va il risultato dell'operazione

$$30000 - 256 \cdot \text{INT}(30000/256)$$

mentre nel secondo quello dell'operazione



INT(30000/256)

Nel primo, cioè, sono contenute le unità, fino a 255. Nel secondo, invece, ci sono le "duecentocinquantesime"; ogni unità del secondo codice vale cioè 256 unità del primo. Questo sistema è da ricordare, perché è il modo in cui vengono tenuti in memoria tutti i numeri a due byte.

Le operazioni successive della routine riguardano il solito trasferimento del valore che ci interessa leggere sullo schermo in bc o in hl, a seconda della macchina. Ricordate a questo proposito che un'operazione tipo

ld bc,a

NON ESISTE, non essendo possibile trasferire in una coppia il valore di un singolo registro. È necessaria perciò la doppia operazione

ld b,0
ld c,a

Proviamo adesso tutto l'inverso. Carichiamo la routine

ld a,30
ld (30000), a
ret

e facciamola eseguire. Se daremo il solito

PRINT USR

sullo schermo apparirà un valore a caso, visto che non abbiamo utilizzato bc (o hl per lo ZX81). Ora, con

PRINT PEEK 30000

andiamo a leggere il contenuto di questa locazione, e vedremo che effettivamente ora è 30.

Qualcosa di più divertente

Un esempio più divertente? (per lo Spectrum)
Provate a far "girare" questo:

```
ld a,156          62 156
ld (22528),a     50  0 88
ret              201
```

dando questa volta

RANDOMIZE USR ...

per non far apparire sullo schermo il valore di bc, che non ci interessa.

Cos'è successo?

Il primo carattere improvvisamente ha iniziato a lampeggiare da rosso a verde. Perché?

Cosa ha a che fare la locazione 22528, che abbiamo modificato, con lo schermo?

Lo Spectrum conserva le informazioni

relative ai cosiddetti "attributi" dello schermo (colore di sfondo, di carattere, luminosità, lampeggio di ciascun carattere) in memoria a partire dall'indirizzo 22528. Ogni byte contiene le informazioni relative ad un carattere (dal primo all'ultimo della prima riga, poi dal primo all'ultimo della seconda e così via). Il modo in cui queste sono codificate è:

bit 7 se è uguale a 1, indica lampeggio (FLASH 1), se è invece a 0, carattere stabile (FLASH 0)

bit 6 se è uguale a 1, indica alta luminosità (BRIGHT 1), se è invece a 0, luminosità normale (BRIGHT 0)

bit 3,4,5 contengono, in forma binaria, il codice del colore di sfondo (PAPER). Così il blu (codice 1) sarà indicato con 001, il giallo (codice 6) con 110, il nero (codice 0) con 000.

bit 0,1,2 contengono, sempre in forma binaria, il codice del colore del carattere (INK).

Valgono perciò le stesse regole spiegate per i colori di sfondo.

Il valore risultante del byte dipenderà quindi dall'insieme di tutte queste informazioni, che assumeranno un "peso" diverso a seconda della loro posizione. Un bit, infatti, con la sua variazione modifica il byte che lo contiene in misura diversa, a seconda della sua posizione:

bit 0	se a 0: 0	se a 1: 1
bit 1	se a 0: 0	se a 1: 2
bit 2	se a 0: 0	se a 1: 4
bit 3	se a 0: 0	se a 1: 8
bit 4	se a 0: 0	se a 1: 16
bit 5	se a 0: 0	se a 1: 32
bit 6	se a 0: 0	se a 1: 64
bit 7	se a 0: 0	se a 1: 128

Tutto si può riassumere nella formula:

$$V=S*2^N$$

dove V è il valore che il bit rappresenta, S lo stato (1 o 0) del bit stesso e N la sua posizione nel byte (da 0 a 7).

Nel nostro caso, abbiamo:

lampeggio : bit 7 a 1=128
luminosità normale : bit 6 a 0= 0
colore sfondo 3 (rosso) : bit 5 a 0= 0
: bit 4 a 1= 16
: bit 3 a 1= 8
colore car. 4 (verde) : bit 2 a 1= 4
: bit 1 a 0= 0
: bit 0 a 0= 0
156

che è il valore che potremmo ottenere sullo schermo con

PRINT BIN 10011100

Con la routine che abbiamo appena visto siamo andati a modificare gli attributi del primo carattere della prima riga, trasformandolo appunto in verde su sfondo rosso e lampeggiante. Se ora facciamo comparire, con

PRINT AT 0,0; "***"

un carattere qualsiasi in quella posizione, e poi diamo nuovamente

RANDOMIZE USR....

il carattere è sempre lì, ma è stato trasformato in verde e rosso lampeggiante.

Non esiste, in BASIC, alcun comando che produca questo effetto (modifica degli attributi senza toccare il carattere), e siamo quindi in presenza di uno di quei casi in cui è necessario l'uso del linguaggio macchina.

Per gli "Zetaottantunisti" abbiamo invece qualcosa di un po' diverso.

In questo computer i programmi BASIC vengono conservati in memoria sempre a partire dall'indirizzo 16509. Se introduciamo la linea

1 REM AAAAA

questa andrà quindi a finire in memoria nelle locazioni da 16509 in su (avendo il numero 1 sarà sempre la prima, anche se in memoria ci sono già altre linee di programma).

Sapendo che il numero di linea occupa due byte (sempre, indipendentemente dal suo valore), che altri due sono occupati dal numero che indica la lunghezza totale della linea (non compare mai nei listati, ma c'è sempre in memoria) ed infine un byte è occupato dal codice dell'istruzione REM possiamo essere certi che la prima A occuperà l'indirizzo 16514 (16509+5).

La routine

```
ld a, 38          62  38
ld (16514),a     50 130 64
ret              201
```

modifica proprio questa locazione, introducendovi il valore del codice della lettera B.

Dopo aver introdotto la linea BASIC che abbiamo visto caricate questa routine con il solito sistema e fatela eseguire. Anche qui è più adatto un

RAND USR

visto che non ci interessa il contenuto di bc, che non usiamo.

Al termine diamo

LIST

e leggeremo

1 REM BAAAA

a prova dell'avvenuta modifica. Sostituendo, nella routine, l'indirizzo 16514 con 16515, e facendola eseguire di nuovo, otterremo la modifica della A successiva.

Ancora un passo avanti

Questo ci porta ad una considerazione:

volendo agire su più indirizzi diversi di memoria, per esempio successivi, con le operazioni viste finora saremmo costretti ad usare

```
ld a, <valore >
ld (indirizzo),a
ret
```

modificando ogni volta l'indirizzo, per poi far eseguire nuovamente la routine. In alternativa potremmo scrivere una lunga routine, tipo

```
ld a, <valore >
ld (indirizzo 1),a
ld (indirizzo 2),a
```

```
ld (indirizzo N),a
ret
```

con tante

```
ld (indirizzo),a
```

quante sono le locazioni da modificare.

Per fortuna esiste un metodo molto più semplice e razionale; osservate questa mnemonica:

```
ld (de),a
```

il suo significato è:

carica la locazione di memoria indicata dal valore di de con a.

Perciò

```
ld de, <indirizzo >
ld a, <valore >
ld (de),a
```

ha la stessa funzione di

```
ld a, <valore >
ld (indirizzo),a
```

Sento già le vostre urla: certo, nel primo caso per fare la stessa cosa abbiamo dovuto usare un'operazione in più. Infatti, non vale la pena di ricorrere a questo sistema per operare su di una singola locazione di memoria.

Osservate però cosa succede quando occorre operare su più locazioni successive: con il vecchio metodo

```
ld a, <valore >
ld (indirizzo 1),a
```

```
ld (indirizzo N) ,a
```

Come caricare una routine in linguaggio macchina

La prima risposta, fin troppo ovvia, è quella di procurarvi un programma assembler e di usarlo; se non lo possedete o preferite lavorare col linguaggio macchina, restando nel più sicuro e comodo ambiente BASIC, usate i seguenti programmi.

Spectrum

```
10 CLEAR 29999
20 LET x=30000
30 INPUT a
40 POKE x,a
50 PRINT x, PEEK x
60 LET x=x+1
70 GO TO 30
```

Il valore a cui viene abbassata RAMTOP, con CLEAR, è adatto per tutti gli Spectrum.

ZX81

Per lo ZX81 la questione è leggermente più complicata: innanzitutto occorre modificare RAMTOP, con

```
10 POKE 16388,93
20 POKE 16389,191
```

dopo di che date NEW. Ora potete utilizzare un programma simile a quello per lo Spectrum:

```
10 LET X=24000
20 INPUT A
30 POKE X,A
40 PRINT X, PEEK X
50 LET X=X+1
60 GO TO 20
```

Per attivare le routine introdotte dovrete dare, a seconda dei casi, RANDOMIZEUSR (o PRINTUSR) 30000 per lo Spectrum, e RANDUSR (o PRINTUSR) 24000 per lo ZX81.

mentre con il nuovo metodo

```
ld de, <indirizzo 1 >
ld a, <valore >
ld (de),a
inc de
ld (de),a
inc de
.
.
ld (de),a
```

Le vostre urla sono sempre più acute: così per ogni caricamento sono necessarie due operazioni, invece di una!

FOR....NEXT? djnz!

E' vero; infatti non è questo il modo corretto di procedere. Comunque abbiamo visto come sia possibile incrementare l'indirizzo semplicemente con

```
inc de
```

L'ideale sarebbe un bel ciclo, proprio come si fa in BASIC con

```
FOR....
NEXT....
```

da ripetere tante volte quanti sono gli indirizzi di memoria da modificare. E' possibile in linguaggio macchina? Certo!

```
djnz <distanza >
```

fa proprio questo.

Questa mnemonica sta per Decrement b, Jump if Not Zero, cioè decrementa b e salta, se non è a zero.

Ogni volta che questa operazione viene eseguita, infatti, il registro b è diminuito di 1. Se il suo valore è ancora maggiore di 0, avviene un salto, la cui destinazione dipende dal valore di <distanza >. Nel caso invece che b valga 0, l'esecuzione passa al codice operativo successivo.

In pratica avviene l'equivalente di

```
FOR b= <val. iniziale > TO 0 STEP -1
```

```
NEXT b
```

oppure, se preferite,

```
LET b=b-1
IF b>0 THEN GO TO ....
```

In pratica, b è il contatore del ciclo e procede sempre a ritroso, terminando a zero.

Proviamo a riscrivere la nostra routine, utilizzando questa operazione:

```
ld b, <val. iniziale>
ld (de), <indirizzo 1>
ld a, <valore>
CICLO ld (de),a
inc de
djnz CICLO
ret
```

Il ciclo verrà percorso tante volte quante occorreranno a b per giungere a 0 da <val. iniziale> da cui era partita. Ogni volta avverrà un salto indietro fino a CICLO, e la sequenza caricamento-incremento sarà ripetuta.

Nel momento in cui b sarà trovata uguale a 0 l'esecuzione passerà alla successiva ret (in questo caso de sarà stata incrementata un'ultima volta, inutilmente).

Ci resta da capire come va calcolato il valore <distanza> per ottenere un salto verso un punto preciso. Non è infatti possibile, in realtà, usare l'etichetta CICLO e pensare che djnz salti proprio fin lì.

Ciò che <distanza> deve contenere è:

$256-(N+2)$

dove N è il numero di byte che separano il codice operativo di djnz dal codice operativo dell'operazione a cui si vuole saltare.

Sembra un po' contorto, non è vero? Cerchiamo di chiarire il perché di tante complicazioni.

- N va sottratto a 256 per indicare un salto all'indietro. E' possibile saltare all'indietro o in avanti di non più di 127 byte. Infatti, valori da 0 a 127 indicano salti in avanti, mentre valori da 128 a 255 salti indietro.

- Quel 2 che va aggiunto serve a compensare il fatto che come punto "zero" per i salti è considerato il codice dell'operazione successiva a djnz, e quindi 2 byte dopo di essa.

L'operazione

```
djnz 0
```

non ha quindi alcun effetto, mentre

```
djnz 255
```

provoca un salto alla stessa djnz.

Proviamo adesso con un esempio, che meglio di qualsiasi spiegazione chiarirà le cose. Provate a contare i byte ed a calcolare il valore da dare a <distanza>, in modo da impadronirvi della tec-

nica.

```
ld de,0      17  0 0
ld b,10     6  10
inc de      19
CICLO djnz CICLO 16 253
ld b,d      66
ld c,e      75
ret         201
```

per lo Spectrum, oppure

```
ld hl,0     33  0 0
ld b,10     6  10
CICLO inc hl 35
djnz CICLO 16 253
ret         201
```

per lo ZX81.

In entrambi i casi la formula dà

$256-(1+2)=253$

trattandosi di un salto al codice precedente.

Per far eseguire le routine usate

PRINT USR....

in modo da verificare che de o hl, a seconda del caso, è stata realmente incrementata 10 volte.

Come vedete, nel caso dello Spectrum c'è qualche istruzione in più, non potendosi usare bc direttamente come è stato fatto per lo ZX81 con hl, visto che b ha funzioni di contatore.

E per finire....

Dopo tanta teoria chiudiamo con due esempi veramente spettacolari, in particolare modo quello per lo Spectrum.

```
ZX81:
ld de,16514  17 130 64
ld b,10     6  26
ld a,28     62  28
ld (de),a   18
CICLO inc de 19
inc a      60
djnz CICLO 16 251
ret         201
```

da usare dopo aver introdotto la linea BASIC

```
1 REM ABCDEFGHIJ
```

che produrrà la linea

```
1 REM 0123456789
```

Per lo Spectrum

```
CICLO ld de,22528  17  0 88
ld b,64         6  64
ld a,156       62 156
ld (de),a      18
CICLO inc de    19
djnz CICLO     16 252
ret            201
```

da usare dopo aver fatto stampare sullo

schermo, con

```
10 FOR i=1 TO 64
20 PRINT "***";
30 NEXT i
```

due righe di asterischi.

Cosa ve ne pare?

Questo per darvi un'idea di cosa si possa ottenere con poche semplici operazioni in linguaggio macchina.

Cercate di ripercorrerne il funzionamento, in modo da comprenderne la meccanica, e arrivederci alla prossima puntata, in cui estenderemo le nostre conoscenze, e vedremo cose ben più fantastiche. A presto!

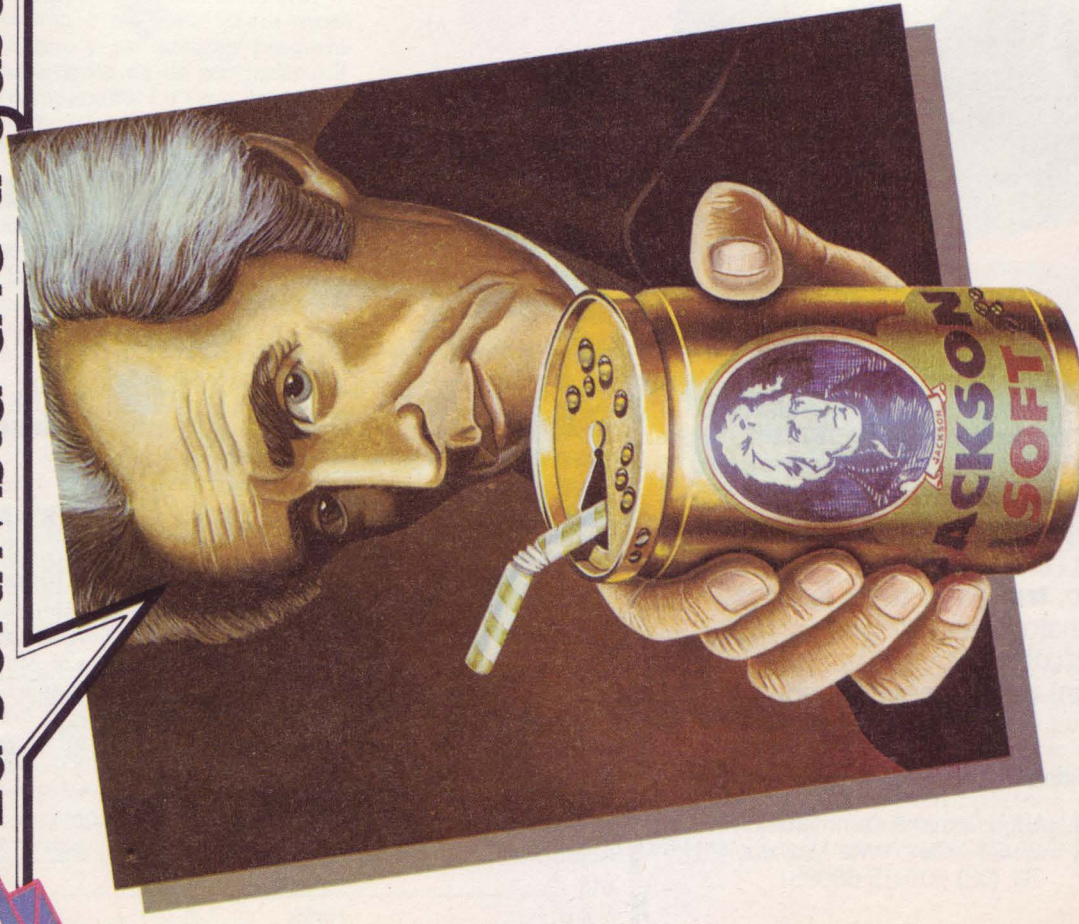
Fig. 1 - disposizione all'interno della memoria dello Spectrum degli attributi per il video.

fig. 2 - meccanica di esecuzione dei due tipi di caricamento di un indirizzo di memoria con il contenuto di un registro.

La cassetta allegata a questo numero di SuperSinc contiene in omaggio, un programma realizzato per lo Spectrum 48K. «Il mod. 740 con lo Spectrum» di Girolamo Potestà.

FINALMENTE!

La Softrivista che ti gasa!



**QUALCOSA DI SUPER, DI INEDITO,
DI IRRESISTIBILE**

**IL VERO GIOCO
COMINCIA ADESSO**

È IN EDICOLA JACKSON SOFT SERIE ORO

La softrivista con i giochi esclusivi per Commodore 64 e Spectrum 48K importati dall'Inghilterra, mai presentati in Italia. Una sfida Jackson al già visto, al già fatto, al... già registrato.

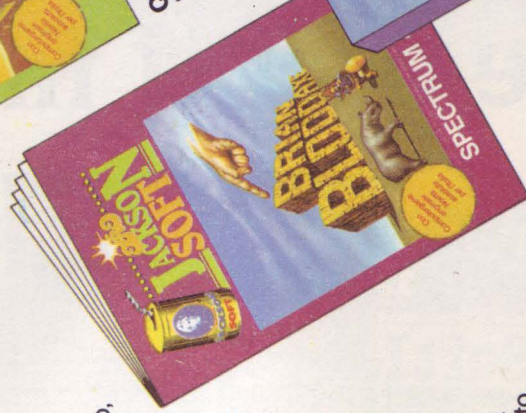


Ogni mese in edicola

- la softrivista con:
 - 6 pagine di commento al supergame presentato;
 - 4 pagine di rubriche;
 - 22 pagine di listati scelti tra il meglio di quanto pubblicato in fatto di grafica, giochi, utility.
- una cassetta originale con il SUPERGAME del mese



QUO WADIS per C 64
Strategia, avventura, logica, mistero e ingenuità. Compone il tuo ingegno nel super gioco di guerra. Insegna a regnare sul mondo del drabino. Poi tu!



BRAN BLODAKE per Spectrum 48K
Un'epica guerra che ti condurrà in pieno 2° secolo. Alle conquiste del grande impero romano. Forza, astuzia, strategia e... dalla corona sarda di Monty Python.



**GRUPPO
EDITORIALE
JACKSON**

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.

DISTRIBUZIONE: un anello strategico tra fornitori e utenti finali.



Soft Dealer: l'unica rivista dedicata al mondo della distribuzione di home e personal computer, software e libri.

Soft Dealer: è il primo mensile italiano per chi vende home e personal computer, software, libri e accessori di informatica. Uno strumento indispensabile per computershop, librerie e cartolerie. Su **Soft Dealer**, inviato solo in abbonamento, troverete tutte le notizie più utili, recensioni, test, prove di software, articoli di marketing, classifiche dei programmi e dei libri più venduti in Italia.

SOFT DEALER è un'idea

PROGRAM
editrice

Uffici Tecnici e Commerciali:
20125 Milano - Via M. Gioia, 114
Tel. (02) 603415-680851

ABBONATI A **Soft Dealer**: Un modesto investimento per essere informati sul futuro

Ritagliare e spedire in busta chiusa a:

PROGRAM EDITRICE, Via M. Gioia 114, 20125 MILANO

- Desidero sottoscrivere un abbonamento annuo (10 numeri L. 25.000) alla rivista SOFT DEALER
- Allego assegno di L. 25.000 intestato a PROGRAM EDITRICE n° Banca
- Ho effettuato pagamento tramite vaglia postale.
- Ho effettuato versamento su ccp n. 34510206 intestato a Program-Milano e allego fotocopia ricevuta.

NOME

COGNOME

AZIENDA

VIA N°

C.A.P. CITTÀ



RANDOMIZE

Allacciamo le stringhe

di Carlo Panzalis

La gestione stringhe del BASIC Sinclair: non standard, ma semplice, elegante e potente.

In questo articolo intendiamo occuparci di un argomento spesso, ed a torto, trascurato: le stringhe e le operazioni che su di esse possono compiersi. Il concetto di stringa, elementarmente intesa come sequenza di caratteri alfanumerici (lettere, numeri, simboli, talvolta caratteri di controllo) racchiusa tra virgolette, spesso risulta ostico all'utente che tenta il primo approccio con la programmazione BASIC, e non di rado le conseguenze di tale prima incomprendimento si trascinano avanti nel tempo, provocando una sottoutilizzazione di questi versatili e semplici strumenti di lavoro.

In realtà non è possibile fare a meno delle stringhe in un qualsiasi programma caratterizzato da un pur minimo livello di complessità. La semplice necessità di stampare sul monitor il nome del programma o una qualunque richiesta rivolta all'utente comportano la composizione e la gestione di una stringa. Tuttavia molto spesso lo strumento stringa non viene maneggiato sfruttando la grande flessibilità che lo caratterizza. È invece opportuno sottolineare che, grazie ad esso, si possono ottenere risultati altrimenti difficilmente realizzabili, interessanti effetti grafici (soprattutto per quanto riguarda l'animazione o la rapida gestione di caratteri grafici) ed altro ancora.

È senz'altro opportuno rimandare il lettore che ha meno confidenza con l'argomento qui trattato al capitolo del manuale Spectrum che ne illustra i concetti fondamentali: non gioverebbero infatti alla continuità del discorso continui richiami ad aspetti elementari. Ad ogni buon conto, in queste stesse pagine allegiamo l'elenco delle principali funzioni che hanno per argomento una stringa, indicandone formato e scopi.

Le operazioni sulle stringhe possono essere ricondotte a tre tipologie fondamentali: la comparazione, la concatenazione, lo slicing.



La comparazione

Le operazioni di comparazione si svolgono su due o più stringhe. La comparazione richiede la utilizzazione degli operatori logici di relazione (=, <, >, < ecc.), sfruttando i quali è possibile confrontare le stringhe sulla base dei codici dei caratteri costituenti le stringhe stesse. Come è noto, ad ogni carattere alfanumerico corrisponde un codice (ASCII): così alla lettera A maiuscola corrisponde il numero di codice 65, alla m minuscola il codice 109 e così via. Comparando le stringhe è dunque possibile organizzarle secondo il consueto ordine alfabetico (ascendente o discendente) o secondo un dato ordine numerico (crescente o decrescente). Tuttavia va sottolineato che, a causa del parametro di confronto utilizzato (il codice

ASCII), possono talvolta manifestarsi alcuni inconvenienti. Infatti il codice della lettera "Z" è minore di quello di "a" o, ancora, i codici delle dieci cifre sono inferiori ai codici di qualunque lettera. È evidente che in casi particolari, tenendo presente queste ultime considerazioni, è comunque possibile approntare procedure rivolte ad evitare questi piccoli inconvenienti.

Segnaliamo che la funzione CODE applicata ad una stringa ritorna il codice del simbolo con cui la sequenza di caratteri ha inizio (salvo specificarne uno diverso); mentre la funzione CHR\$ applicata ad un simbolo ne ritorna il codice ASCII (sia detto per inciso: nel set di caratteri Spectrum ve ne sono due, la sterlina e il copyright, che non sono previsti nello standard ASCII).

Concatenazioni

Un secondo tipo di operazioni riguarda la concatenazione di stringhe.

L'operatore per eccellenza in questo caso è rappresentato dal simbolo "+", che ha ovviamente lo scopo di fondere due o più stringhe in una sola. Lo statement `LET a$="SUPER"+"SINC"` unisce le due stringhe, attribuendo ad `a$` il valore "SUPERSINC". Attenzione, però: l'addizione tra stringhe è profondamente differente dalla somma tra numeri.

Così

`LET a=33+44` attribuisce alla variabile `a` il valore 77

mentre

`LET a$="33"+"44"` attribuisce alla variabile `a$` il valore "3344".

Una parola va spesa anche su un operatore che sembra concatenare le stringhe, ma che in realtà non arriva mai a fonderle insieme, come visto precedentemente: ci riferiamo al simbolo ";" usato nell'ambito di una istruzione `PRINT`. Il fatto di stampare sul video due stringhe una di seguito all'altra, ricorrendo a questo operatore (ad esempio `PRINT "SUPER";"SINC"`), non significa averle concatenate: infatti le due stringhe continuano ad avere vita autonoma e possono essere gestite separatamente. Questa procedura è naturalmente valida ogniqualvolta la concatenazione non è richiesta durante tutto lo svolgimento del programma, ma serve solo in un preciso istante per visualizzare date informazioni.

Lo slicing

Infine, la più classica operazione sulle stringhe è detta slicing (da to slice, affettare, dividere in parti...): con tale termine si intende la possibilità di estrarre da una stringa data una o più sottostringhe, sulla base della posizione occupata da ciascun carattere nella stringa "madre".

Normalmente, in BASIC le tipiche funzioni per lo slicing sono `LEFT$`, `MID$` e `RIGHT$`. Il formato di tali funzioni è:

funzione (variabile stringa, indice di partenza, indice d'arrivo).

Ad esempio, data una variabile stringa `a$` inizializzata "SUPERSINC", lo statement `MID$(a$,3,7)` estrae la sottostringa "PERSI". Per semplificare il tutto, `LEFT$` e `RIGHT$`, che si riferiscono, come suggerisce il nome, all'estrazione di una sottostringa dalla sinistra (inizio) o dalla destra (fine) della stringa originale, operano considerando rispettivamente come sottintesi il carattere di partenza o di arrivo: è quindi sufficiente specificare il numero di caratteri che si intendono considerare per attivare queste due funzioni.

A ben considerare, questi statement appaiono a prima vista sgradevoli e scarsamente "user friendly", poco indirizzati all'utente. A motivo di ciò può essere senz'altro apprezzata la nuova forma di tali funzioni implementata sullo Spectrum: al triplice comando è stato infatti sostituito un unico operatore, il cui significato è palese fin dal primo colpo d'occhio. La funzione in questione assume la forma:

variabile stringa (indice di partenza TO indice d'arrivo).

Tipica forma è dunque `PRINT a$(3 TO 7)`.

Se si intende considerare una stringa dall'inizio o sino alla fine, è possibile omettere i rispettivi indici. Così, per togliere il primo carattere da una stringa `a$` è possibile formulare:

`LET a$=a$(2 TO);`

analogamente si procede per espellere dalla stringa l'ultimo carattere, senza bisogno di specificare l'indice di partenza.

Di qualsiasi stringa può essere considerato un singolo elemento, facendo riferimento alla sua posizione nella sequenza. Tale possibilità si realizza con le scritture `a$(3)` oppure `a$(3 TO 3)`.

In realtà, con il nuovo operatore nulla muta sostanzialmente rispetto alla formulazione tradizionale, tuttavia vi è un effettivo incremento di leggibilità, soprattutto da parte di chi tenta di "affettare" le stringhe per la prima volta.

Divertiamoci un po'

A questo punto concentreremo la nostra attenzione sulle operazioni di concatenazione e sullo slicing, mostrando come queste due operazioni possano condurre a risultati gradevoli.

Per il primo esempio considereremo un classico effetto tipo "insegna luminosa", intendendo cioè il passaggio sul video di una qualsiasi scritta. Siamo fra l'altro certi che i lettori conoscono bene questo effetto, visto che esso è stato spesso sfruttato in programmi commerciali ad altissima diffusione (ad esempio per le istruzioni di gioco di Manic Miner, Jet Set Willy, ecc.). Alla base del "trucco" sta l'assegnazione ad una variabile stringa della sequenza di caratteri che

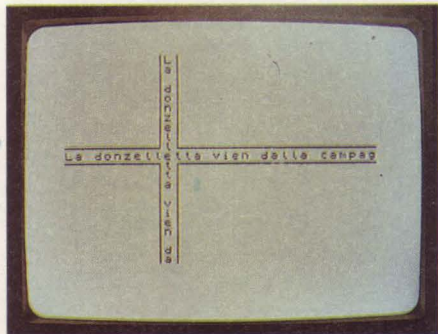
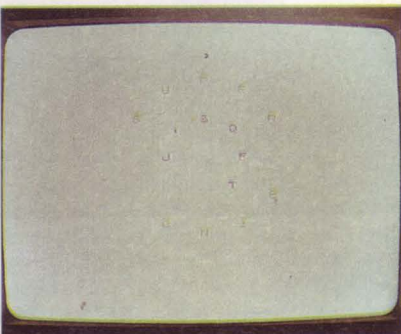
intendiamo rappresentare, quindi la richiesta di stampare sullo schermo i primi 32 caratteri della scritta; su questi caratteri andranno poi ad imprimerli successivamente altri 32 caratteri, incrementando man mano l'indice di partenza e quello di arrivo.

Un esempio di quanto ora descritto è il seguente:

```
10 LET a$ = "Ecco una scritta che
   scrollerà lentamente sul video"
50 FOR f = 1 TO 21
60 LET b$ = a$(f TO f+31)
70 PRINT AT 10,0;b$
80 PAUSE 10
90 NEXT f
```

Della stringa `a$` vengono stampati (sempre nella medesima posizione!) 32 caratteri, da 1 a 32; poi da 2 a 33 e così via. Il problema è che, come potete notare lanciando il programma, il ciclo non è completo: in particolare vi è una parte della scritta che non esce mai dallo schermo e ve n'è una che non rientra mai. Aggiungere una linea `100 GO TO 50` non serve allo scopo, poiché è evidente che la scritta non scorre, neanche in questo caso, secondo il modo da noi richiesto. È invece necessario (anche se non è il solo procedimento possibile) inserire una linea così fatta: `20 LET a$ = a$ + a$`; e quindi modificare il secondo indice del loop `FOR..NEXT`, sostituendo al valore 21 il valore che rappresenta la lunghezza della stringa `a$` originale (quindi `50 FOR f = 1 TO 52`). A questo punto la linea 100 come sopra indicata realizzerà la iterazione, ma, con questi nuovi valori, secondo le modalità da noi desiderate.

Se vi è chiaro il meccanismo ora descritto, non dovrete incontrare soverchie difficoltà a comprendere come funzionano i due brevi programmi "INCROCIO" e "GIROTONDO" qui pubblicati. Dall'esame del primo listato dovrebbe emergere evidente una considerazione: la realizzazione di scritte in verticale è decisamente meno agevole rispetto alla impressione sul video di scritte orizzontali! Il listato vi richiede di inserire una sequenza di caratteri (al massimo 80, ma non si tratta di un limite invalicabile), dopo di che provvede a gestire la stringa in modo da realizzare un piacevole effetto grafico. Notate che il loop più esterno, quello che utilizza la variabile di controllo `w`, serve soltanto a con-



sentire la ripetizione dell'output un numero definito di volte. Volendo, è possibile sostituirlo con un GO TO 110, facendo così girare le scritte all'infinito, magari fino a che l'utente non prema un tasto (ponendo IF INKEY\$ < > " THEN GO TO...., ad esempio).

Il secondo listato è di ancor più facile comprensione: sfruttando le funzioni trigonometriche SIN e COS si identificano delle posizioni di stampa, quindi attraverso lo slicing carattere per carattere vi si stampano le singole lettere che compongono la stringa (anzi le stringhe, visto che sono due!). In particolare le linee 10-60 sono di inizializzazione, le linee 100-250 determinano le coordinate di stampa ed infine il loop 300-400 si occupa della stampa su schermo delle stringhe, nonché della loro animazione.

Il terzo listato (Da 3 a tre) si occupa della traduzione dei numeri-cifra in numeri-lettere. Una volta avviato il programma richiede all'utente di inserire un numero qualsiasi (al di sotto del milione): dopo qualche istante sullo schermo apparirà il numero stesso tradotto in lettere. Il cuore del programma è costituito da una serie di linee DATA, contenenti le parole-chiave per la traduzione, che vengono lette in modo da essere associate al numero da tradurre. Il numero inserito dall'utente viene tradotto in stringa, successivamente viene "affettato" con lo slicing per individuare la posizione di ogni cifra che lo costituisce e il corrispondente valore. Prendiamo qui spunto dalla scarsa utilità pratica di questo programma per sottolineare che tutti i listati qui presentati non sono finalizzati a risolvere problemi concreti dell'utente, ma solo a mostrare esempi divertenti di gestione delle stringhe. L'obiettivo è dunque palesemente quello di incuriosire e far riflettere chi ha meno familiarità con il tema trattato, cercando di suggerire spunti per lavori che abbiano invece l'utilità concreta che manca a questi. In tal senso i programmi presentati sono decisamente migliorabili sotto numerosi profili ed in taluni casi sottostanno a limitazioni di varia natura (massima lunghezza di certe stringhe, massimo valore dei numeri da tradurre), che sono imposte solo dalla reale irrilevanza del vincolo stesso: è cioè chiaro che non esistono difficoltà a superare tali vincoli, semplicemente non è interessante farlo in questa sede. Infine tenete presente che, nell'ambito del discorso appena fatto, per non ap-

pesantire la struttura del listato non sono state approntate sicure sugli INPUT particolarmente efficienti: prestate dunque una certa attenzione nell'inserire dati da tastiera.

Infine esaminiamo brevemente il quarto listato, vale a dire Stringhe e UDG.

Come già il nome suggerisce, il programma si avvale di stringhe costituite esclusivamente da caratteri grafici definiti dall'utente, e ciò allo scopo di realizzare un'animazione grafica con poca fatica. Il listato è talmente semplice e breve da non richiedere alcuna spiegazione particolare: dopo aver definito dei caratteri grafici si inizializzano due stringhe; queste ultime vengono poi impresse alternativamente sullo schermo, e questo flip=flop simula egregiamente il movimento di un gruppo di animali. Può forse essere utile svolgere qualche considerazione sul tema. Innanzitutto non è necessario che le stringhe siano composte da caratteri grafici identici fra loro: ad esempio, l'effetto di movimento può migliorare, sostituendo nella stringa a\$ ad un paio di coppie di UDG "AB" le coppie "CD" e viceversa. In secondo luogo non è necessario che le stringhe siano lunghe sempre 32 caratteri, anzi spesso è meglio che esse siano più brevi. Supponiamo ad esempio di voler "incorniciare" il branco di animali in movimento con il disegno di un fine-

strino, simulando cioè la visione che si potrebbe avere da un treno in corsa o da una Land Rover impegnata in una battuta di caccia: in tal senso si rende opportuno accorciare le stringhe, onde evitare che agli estremi l'orda di animali cancelli il disegno del finestrino stesso. Infine considerate che più stringhe possono essere contemporaneamente mosse sullo schermo in direzioni diverse e seguendo differenti modalità.

Concludendo...

Chiodiamo questa chiacchierata sulle stringhe sottolineandone ancora una volta versatilità ed importanza. I brevi programmi considerati nell'articolo hanno indicato alcune possibili applicazioni sulle stringhe: si tratta comunque solo di esempi, esempi che non esauriscono certamente l'argomento. Basti pensare alla possibilità di utilizzare stringhe per costituire archivi, per sostituire dove può risultare conveniente linee DATA (o dove queste ultime non possono essere usate; vedere lo ZX81), oppure per risolvere problemi di scarsità di memoria disponibile. Al di là di questi od altri possibili esempi è evidente che tali strumenti rappresentano elementi centrali della programmazione BASIC, ed in tal senso non possono e non devono essere trascurati.

Alcune funzioni applicate alle stringhe

LEN	ritorna la lunghezza di una stringa. Caso particolare: i comandi BASIC "tokenizzati". Se, ad esempio, ponete A\$ = "STOP", LEN A\$ è 1 (in questo senso potete anche salvare programmi con nomi più lunghi di 10 lettere: ad esempio: "WAIT THEN STOP TAPE!")
STR\$	converte i numeri in stringhe
VAL	converte le stringhe in numeri. STR\$ e VAL vengono spesso usati in coppia, ad esempio per valutare la correttezza di un INPUT numerico o, in abbinamento a LEN, per determinare la posizione di stampa di tabelle di numeri
CHR\$	applicata ad un numero ritorna il carattere cui corrisponde quel codice. Spesso CHR\$ viene impiegata per ottenere risultati particolari (ad esempio, CHR\$ 8 sposta il cursore indietro di una posizione) o per stampare caratteri grafici (ad esempio, PRINT CHR\$ 134 stampa il simbolo grafico posto sul tasto 6)
CODE	applicato ad una stringa ritorna il codice del primo carattere o di altro elemento della stringa specificato. Anche CODE e CHR\$ vengono spesso utilizzati per controllare gli INPUT
(indice TO indice)	permette di estrarre sottostringhe da una stringa data. In diversi casi uno degli indici può essere tralasciato (o sottinteso)

Inoltre operazioni sulle stringhe possono essere effettuate con gli operatori logici di relazione (maggiore, minore, uguale, diverso...) e con il simbolo di addizione (+).

Incrocio

versione per ZX Spectrum 16K/48K

```
1 REM *****
2 REM INCROCIO DI STRINGHE
3 REM *****
10 DIM a$(96): DIM b$(160)
```

```
20 PRINT AT 3,0;"INSERISCI UNA STRI
NGA DI MASSIMO80 CARATTERI : "
30 INPUT a$
40 FOR g=1 TO 80: LET b$(g)=a$(g):
NEXT g
50 FOR g=81 TO 160: LET b$(g)=a$(g-
80): NEXT g
```

```

60 FOR g=0 TO 7: PAUSE 5: BORDER g:
  NEXT g: CLS
70 PLOT 0,98: DRAW 77,0: DRAW 0,77:
  PLOT 0,85: DRAW 77,0: DRAW 0,-8
  5
80 PLOT 91,0: DRAW 0,85: DRAW 164,0
  : PLOT 91,175: DRAW 0,-77:
  DRAW 164,0
100 FOR w=1 TO 10
110 FOR j=1 TO 80
120 IF j>50 THEN GO TO 150
130 PRINT AT 10,0;a$(j TO j+31);
140 GO TO 190
150 PRINT AT 10,0;a$(j TO 80);a$(1
  TO j-49);
190 FOR k=0+j TO 21+j
220 PRINT AT k-j,10;b$(k TO k)
300 NEXT k
390 NEXT j
400 NEXT w

```

Girotondo

versione per ZX Spectrum 16K/48K

```

1 REM *****
2 REM *      GIROTONDO      *
3 REM *****
5 CLS : PRINT AT 10,10;"Un attimo.
  .."
10 DIM a(36)
20 DIM b(36)
30 DIM c(32)
40 DIM d(32)
50 LET a$=" SUPER SINC"
60 LET b$=" J.SOFT"
100 FOR m=0 TO 24 STEP 12
110 FOR n=1 TO 12
120 LET a(n+m)=10+8*COS (n/6*PI)
130 LET b(n+m)=16-8*SIN (n/6*PI)
140 NEXT n
150 NEXT m
200 FOR m=0 TO 24 STEP 8
210 FOR n=1 TO 8
220 LET c(n+m)=10+4*COS (n/4*PI)
230 LET d(n+m)=16-4*SIN (n/4*PI)
240 NEXT n
250 NEXT m
260 CLS
300 FOR m=1 TO 24
310 FOR n=m TO m+11
320 PRINT AT a(n),b(n); INK 4;a$(n+1
  -m)
330 NEXT n
350 FOR n=26-m TO 33-m
360 PRINT AT c(n),d(n); INK 3;b$(n-2
  5+m)
370 NEXT n
390 NEXT m
400 GO TO 300

```

Da 3 a tre

versione per ZX Spectrum 16K/48K

```

1 REM *****
2 REM *      DA 3 A TRE      *
3 REM *****
10 INPUT "NUMERO ",m
20 LET m$=STR$(m)
30 LET w=LEN m$: LET o=0
40 IF w>6 THEN GO TO 10
45 IF w<=3 THEN GO TO 90
50 IF w>=4 THEN LET o=2
60 LET b$=m$(1 TO w-3)
70 LET c$=m$(w-2 TO w)
80 GO TO 95
90 LET n=m: GO TO 100
95 LET b=VAL b$: LET n=b: GO SUB 10
  0
96 LET c=VAL c$: LET n=c: GO SUB 10
  0
97 GO TO 9000
100 DIM a$(30,11)
150 FOR j=1 TO 30
160 READ a$(j)
180 NEXT j
600 DATA "zero","uno","due","tre","q
  uattro"
610 DATA "cinque","sei","sette","ott
  o","nove"
620 DATA "zero","dieci","venti","tre
  nta","quaranta"
630 DATA "cinquanta","sessanta","set
  tanta","ottanta","novanta"
640 DATA "zero","undici","dodici","t
  redici","quattordici"
650 DATA "quindici","sedici","dicias
  sette","diciotto","diciannove"
1010 INK 7
1020 LET u$=STR$(n)
1030 LET v=LEN u$
1050 IF v=1 THEN GO TO 3000
1060 IF v=2 THEN GO TO 3500
1070 IF v=3 THEN GO TO 6000
3000 REM
3010 PRINT a$(n+1)
3020 GO TO 9000-o
3500 REM
3510 LET k$=u$(2): LET h$=u$(1)
3520 LET h=VAL h$
3530 LET k=VAL k$
3540 IF k=0 THEN GO TO 3700
3550 IF k<>0 THEN GO TO 4000
3700 PRINT a$(h+11)
3710 GO TO 9000-o
4000 IF h>=1 AND h<2 THEN GO TO 4500
4050 IF h>=2 THEN GO TO 5000
4500 PRINT a$(n+11)
4510 GO TO 9000-o
5000 LET sp=LEN a$(11): FOR t=1 TO sp
  : PRINT a$(h+11,t);
5010 IF a$(h+11,t)=" " THEN PRINT
  CHR$(8);
5030 NEXT t
5040 PRINT a$(k+1)

```

```

5050 GO TO 9000-o
6000 REM
6010 LET x$=u$(1): LET h$=u$(2):
      LET k$=u$(3)
6020 LET x=VAL x$
6025 LET h=VAL h$
6030 LET k=VAL k$
6035 LET y=VAL u$
6040 LET l$=u$(2 TO 3)
6050 LET l=VAL l$
6100 IF x=1 THEN LET a$(2)=" "
6110 IF y=101 OR y=111 THEN GO TO 80
      00
6115 IF x=1 AND k=1 THEN GO TO 8500
6120 IF h=0 AND k=0 THEN GO TO 6200
6140 GO TO 6500
6200 LET sp=LEN a$(11): FOR t=1 TO sp
      : PRINT a$(x+1,t);
6210 IF a$(x+1,t)=" " THEN PRINT
      CHR$ 8;
6230 NEXT t
6240 PRINT "cento"
6250 GO TO 9000-o
6500 IF k=0 AND h<>0 THEN GO TO 65
      02
6501 GO TO 6520
6502 LET a$(11)="": LET sp=LEN a$(13)
      : FOR t=1 TO sp: PRINT a$(x+1,t)
      ;
6503 IF a$(x+1,t)=" " THEN PRINT
      CHR$ 8;
6504 NEXT t
6506 PRINT "cento";a$(h+11)
6510 GO TO 9000-o
6520 IF h=0 AND k=1 THEN GO TO 6522
6521 GO TO 6530
6523 LET sp=LEN a$(13): FOR t=1 TO sp
      : PRINT a$(x+1,t);
6525 IF a$(x+1,t)=" " THEN PRINT
      CHR$ 8;
6526 NEXT t
6527 PRINT "centouno"
6529 GO TO 9000-o
6530 IF h=0 AND k<>0 THEN GO TO 65
      35
6532 GO TO 6550
6535 LET sp=LEN a$(13): FOR t=1 TO sp
      : PRINT a$(x+1,t);
6537 IF a$(x+1,t)=" " THEN PRINT
      CHR$ 8;
6540 NEXT t
6542 PRINT "cento";a$(k+1)
6550 IF h>=1 AND h<2 AND k<>0
      THEN GO TO 6560
6555 GO TO 6600
6560 LET sp=LEN a$(13): FOR t=1 TO sp
      : PRINT a$(x+1,t);
6562 IF a$(x+1,t)=" " THEN PRINT
      CHR$ 8;
6565 NEXT t
6570 PRINT "cento";a$(l+11)
6600 IF h>=2 AND k<>0 THEN GO TO 6
      620
6610 GO TO 9000-o
6620 LET sp=LEN a$(13): FOR t=1 TO sp
      : PRINT a$(x+1,t);

```

```

6625 IF a$(x+1,t)=" " THEN PRINT
      CHR$ 8;
6627 NEXT t
6630 PRINT "cento";
6635 LET sp=LEN a$(13): FOR t=1 TO sp
      : PRINT a$(h+11,t);
6637 IF a$(h+11,t)=" " THEN PRINT
      CHR$ 8;
6640 NEXT t
6645 PRINT a$(k+1)
6700 GO TO 9000-o
8000 IF y=101 THEN PRINT "";"centoun
      o"
8010 IF y=111 THEN PRINT "";"centoun
      dici"
8300 GO TO 9000-o
8500 LET sp=LEN a$(11): PRINT "cento"
      ;: FOR t=1 TO sp: PRINT a$(h+11,
      t);
8510 IF a$(h+11,t)=" " THEN PRINT
      CHR$ 8;
8530 NEXT t
8540 PRINT "uno"
8550 GO TO 9000-o
8998 RESTORE : RETURN
9000 DIM d$(32): DIM l$(32)
9003 FOR r=0 TO 1
9005 FOR i=0 TO 31
9015 PRINT AT 2+r,0+i;SCREEN$(r,i);
9020 LET d$(i+1)=SCREEN$(0,i)
9025 LET l$(i+1)=SCREEN$(1,i)
9050 NEXT i
9055 NEXT r
9060 INK 0: LET cd=0: FOR j=1 TO 31
9065 IF d$(j)="a" AND d$(j+1)="o"
      THEN LET cd=cd+1: GO TO 9290
9070 IF d$(j)="a" AND d$(j+1)="u"
      THEN LET cd=cd+1: GO TO 9290
9080 IF d$(j)="o" AND d$(j+1)="o"
      THEN LET cd=cd+1: GO TO 9290
9090 IF d$(j)="i" AND d$(j+1)="o"
      THEN LET cd=cd+1: GO TO 9290
9100 IF d$(j)="i" AND d$(j+1)="u"
      THEN LET cd=cd+1: GO TO 9290
9110 IF w=4 AND d$(j)="u" AND d$(j+1)
      ="n" THEN PRINT AT 17,j-1;"mill
      e": GO TO 9300
9250 PRINT AT 17,j-1-cd;d$(j);
9290 NEXT j
9292 FOR z=0 TO 31: INK 7: PRINT AT 0
      ,0;SCREEN$(17,z);
9294 IF SCREEN$(17,z)=" " AND w>=4
      THEN GO TO 9298
9296 NEXT z
9297 GO TO 9300
9298 INK 0: PRINT AT 17,z;"mila"
9300 INK 0: LET cd=0: FOR j=1 TO 31
9305 IF l$(j)="a" AND l$(j+1)="o"
      THEN LET cd=cd+1: GO TO 9500
9310 IF l$(j)="a" AND l$(j+1)="u"
      THEN LET cd=cd+1: GO TO 9500
9320 IF l$(j)="o" AND l$(j+1)="o"
      THEN LET cd=cd+1: GO TO 9500
9330 IF l$(j)="i" AND l$(j+1)="o"
      THEN LET cd=cd+1: GO TO 9500

```

SUPERSINC

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA da compilare e spedire in busta chiusa a J.soft - Viale Restelli, 5 - 20124 Milano - Tel. 6888228

Vogliate inviarmi i numeri di SUPERSINC con cassetta

Al prezzo di L. 15.000

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul cc. n° 19445204 intestato a J.soft - Milano

Pagherò in contrassegno al postino al ricevimento

Nome

Cognome

Via

Città

C.A.P.

Prov.

Se richiesta fattura:
Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

BUONO D'ORDINE PER RIVISTE ARRETRATE

Anche se l'ordine riguardasse la sola cassetta questa verrà comunque inviata insieme alla rivista al prezzo qui indicato.

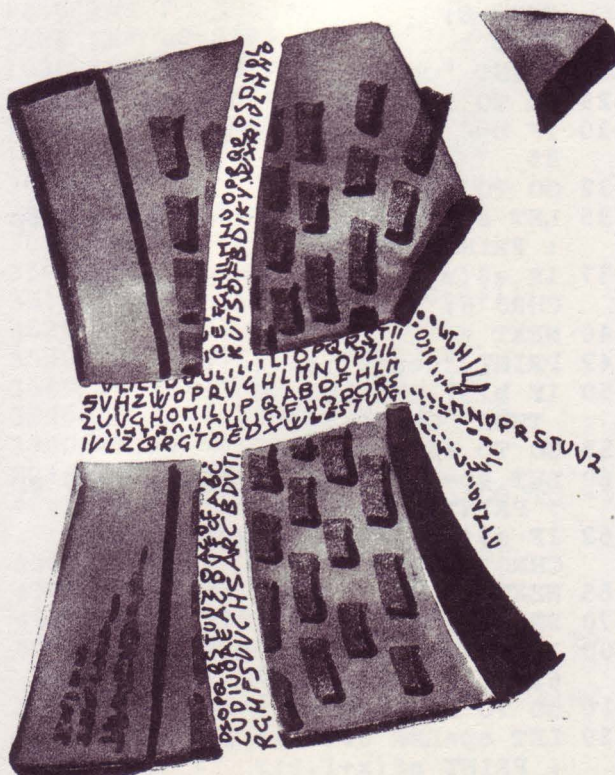
Per ricevere le riviste arretrate di SUPERSINC con cassetta compilate e spedite il coupon qui sotto!

```
9340 IF 1$(j)="i" AND 1$(j+1)="u"
      THEN LET cd=cd+1: GO TO 9500
9350 IF 1$(j)="z" THEN GO TO 9900
9400 PRINT AT 18,j-1-cd;1$(j);
9500 NEXT j
```

Stringhe e UDG

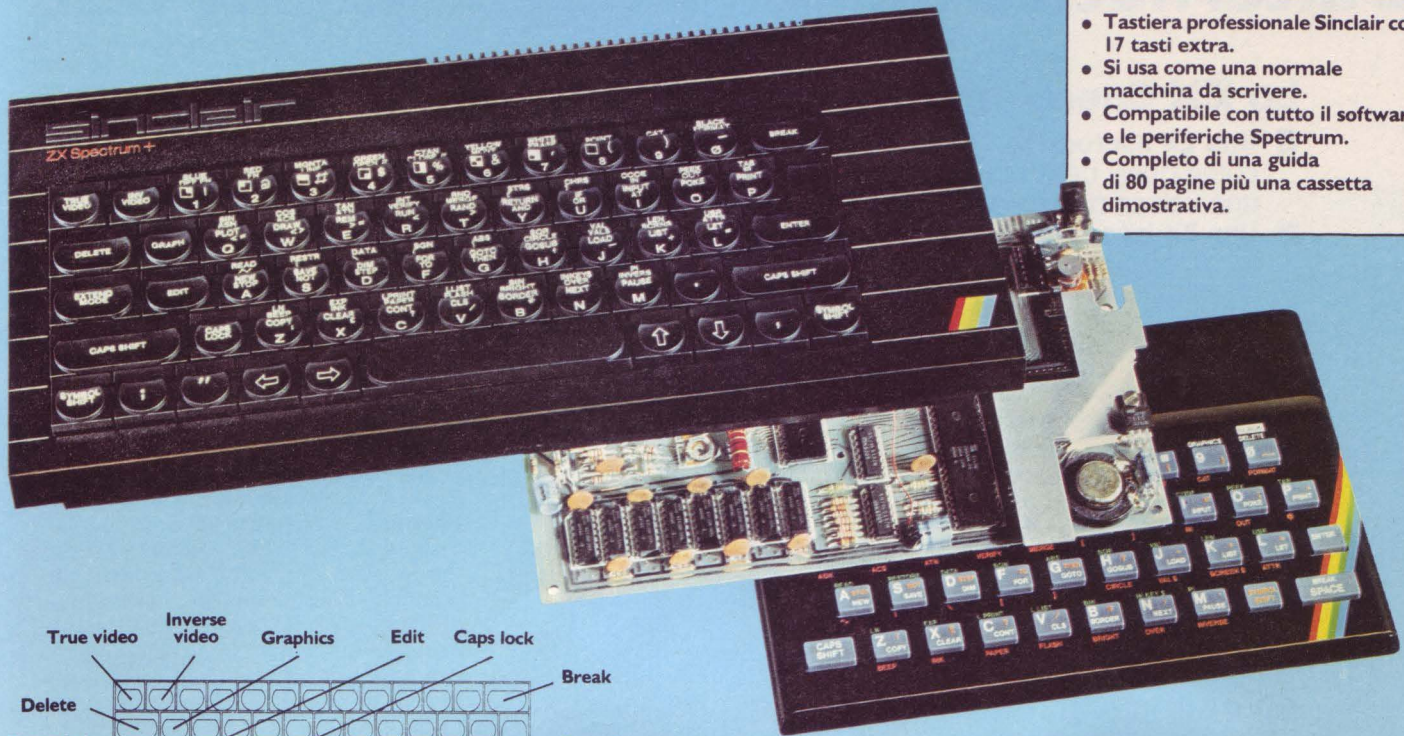
versione per ZX Spectrum 16K/48K

```
1 REM *****
2 REM * STRINGHE & UDG *
3 REM *****
4 REM
10 FOR f=USR "a" TO USR "d"+7
20 READ a
30 POKE f,a
40 NEXT f
50 DATA 192,48,15,7,59,64,128,0
60 DATA 12,30,240,224,220,2,1,0
70 DATA 32,32,63,7,11,8,8,6
80 DATA 12,30,240,224,208,16,16,96
100 REM *****
101 REM * ANIMAZIONE *
102 REM *****
110 CLS
120 LET a$=" AB AB AB AB AB
      AB AB"
130 LET b$=" CD CD CD CD CD
      CD CD"
140 FOR n=1 TO 100
150 LET a$=a$(32 TO )+a$( TO 31)
160 LET b$=b$(32 TO )+b$( TO 31)
170 PRINT AT 7,0;a$
175 PAUSE 4
180 PRINT AT 7,0;b$
185 PAUSE 4
190 NEXT n
```

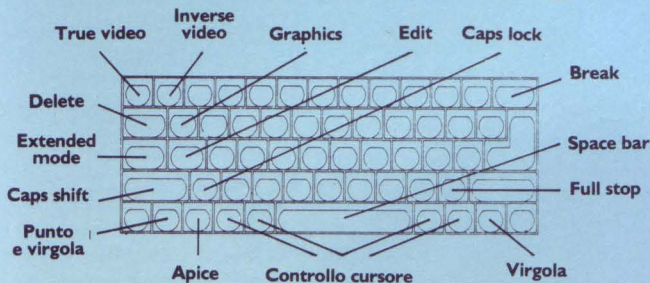


Novità !! Lo Spectrum maggiorato !!!

Trasforma il tuo Spectrum in Spectrum +



- Tastiera professionale Sinclair con 17 tasti extra.
- Si usa come una normale macchina da scrivere.
- Compatibile con tutto il software e le periferiche Spectrum.
- Completo di una guida di 80 pagine più una cassetta dimostrativa.



sinclair

Ecco una novità stimolante per i possessori di Spectrum:
il Kit originale Sinclair che promuove lo Spectrum al grado superiore.

Non si richiede vasta esperienza. Basta saper saldare pochi fili.

Nel Kit sono contenute chiare istruzioni in italiano.

La tastiera dello Spectrum +

Le dimensioni dello "Spectrum +", sono 320x150x40. La tastiera, di tipo dattilografico, ha solidi tasti antiurto. Il suo uso è morbido e ideale per la scrittura al tocco, per il word processing, per i programmi di simulazione e i programmi avanzati.

Vi sono 58 tasti, di cui 17 rappresentano delle novità. I programmatori avranno la gradita sorpresa di trovare i tasti della punteggiatura e, a parte, i tasti "shift", per i grafici e gli "extended modes". Il tasto di reset consente di cancellare un programma dalla memoria del computer senza scollegare l'alimentazione.

Lo Spectrum maggiorato

Naturalmente il computer di grado superiore accetta tutte le periferiche del Sinclair System-Interface I, i Microdrives, eccetera, come pure tutto il software Spectrum.

I nuovi software e le nuove periferiche Spectrum saranno progettati tenendo conto dello Spectrum +, cosicché lo Spectrum accresciuto di grado avrà nuove capacità e potenziale nuovo per il futuro, oltre ad essere stilisticamente fantastico!!!



Il **Kit** contiene le istruzioni per l'assemblaggio, il manuale e la cassetta «demo didattica» in italiano.

REBIT
COMPUTER
A DIVISION OF G.B.C.

In vendita presso i rivenditori specializzati

ZX Spectrum Expansion System

L'alternativa della Sinclair ai floppy disc

Lo ZX Spectrum Expansion System contiene:

- **Uno ZX Microdrive** - Che amplia la possibilità dello ZX Spectrum in quei settori, come quelli della didattica e delle piccole applicazioni gestionali, dove è necessaria una veloce ricerca delle informazioni memorizzate su un supporto magnetico.
- **Una ZX Interface 1** - Indispensabile per il collegamento dello ZX Microdrive. Incorpora una interfaccia RS 232 e un sistema di collegamento in rete locale.
- **Quattro cartucce Microdrive comprendenti un programma di:**
 - Word processor «Tasword Secondo»
 - Masterfile filing system
 - Inventore di giochi
 - Le formiche giganti
- Un programma dimostrativo del Microdrive
- Documentazione per il collegamento, il funzionamento e altre descrizioni tecniche.
- Cavi di collegamento allo ZX NET che può collegare fino a 64 computer ZX Spectrum o QL.



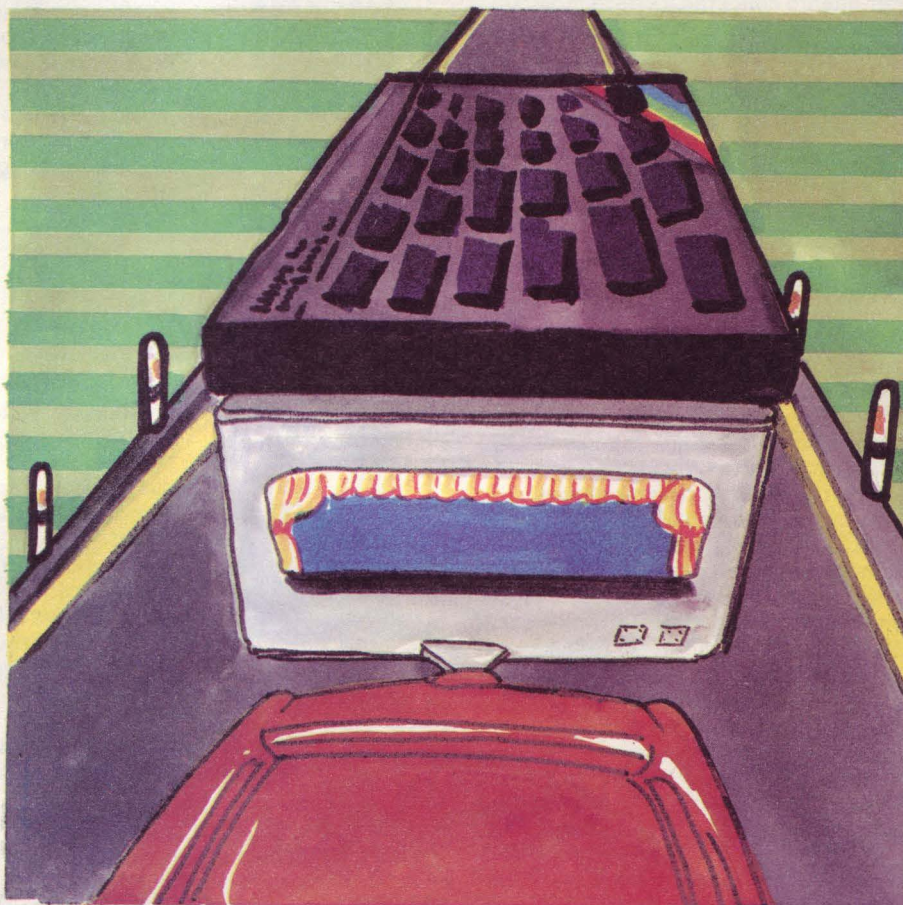
In vendita presso
i rivenditori specializzati

Gita turistica

di **Colin Gooch**
trad. e adatt.
di **Carlo Panzalis**

Un simpatico boardgame dal sapore tipicamente "estivo" per lo Spectrum 48K.

I cosiddetti "giochi da tavolo" stanno vivendo un momento difficile. I dietrologi, studiosi che spendono la loro vita nel tentativo di spiegarci (e spiegarsi) cosa si nasconde dietro fenomeni giudicati irrilevanti dall'uomo della strada (altra categoria socio-psico-filosofica di malcerta identificabilità), affermano che i motivi della crisi debbono essere probabilmente ricercati nella sostanziale staticità evolutiva di tali giochi. In altri termini, tutti i giochi di società sarebbero figli di un unico (e ormai canuto e stanco) progenitore, il Grande Gioco-delloca, che troneggia ed impera su tutti ed ai lati del quale siedono, tronfi e presuntuosi, i suoi vassalli più fedeli: dadi e segnalini. La tesi dominante sembra dunque essere quella secondo cui visto un gioco da tavolo li hai visti tutti! Senza voler fare del darwinismo spicciolo, gli studiosi sembrano affermare che si tratta di giochi che non si evolvono, che non riescono ad adeguarsi al cambiamento ambientale, ed in tal senso sono destinati all'estinzione, seguendo il destino che fu dei goffi e pesanti dinosauri. Francamente, e con buona pace dei dierologi, preferiamo attribuire l'aresponsabilità della crisi a circostanze più palpabili e di minor portata. Innanzitutto è necessario che amici fedeli abbiano voglia e tempo di giocare quando noi abbiamo tempo e voglia di giocare. In secondo luogo va riconosciuto che organizzativamente e fisicamente il gioco da tavolo comporta numerosi problemi, quali, ad esempio, trovare i dadi che non sono stati rimessi a posto l'ultima volta o eliminare quel cartellino degli obiettivi talmente spiegazzato da essere riconosciuto immediatamente da ogni giocatore che sieda abitualmente al nostro tavolo. Comunque, sia che si tratti di cause profonde e irreversibili o dei piccoli intoppi quotidiani, la risposta che il gioco di società ha scelto per la conservazione della specie, cioè per la



sua stessa sopravvivenza, prende sempre più forma: la mutazione genetica! Ecco dunque tabelloni di gioco e dadi, cartellini degli imprevisti e carri armati invadere silenziosamente, ma con determinazione, le RAM dei più disparati home computer, per rientrare "alla grande" nelle nostre case. Se l'amico non viene, si può giocare contro la macchina, provando il sottile brivido della lotta contro un avversario implacabile, ma poco estroso; la ricerca affannosa dei vari pezzi del gioco è ormai un pallido e ridicolo ricordo (purché troviate la cassetta su cui avete registrato il programma!...); le grandi capacità di immagazzinamento ed elaborazione del computer consentono di ritrovarsi in situazioni sempre differenti, al punto che

spesso numerose partite vengono giocate senza aver esplorato e sperimentato tutti i possibili eventi previsti dal programmatore. A questo punto è evidente che "Gita Turistica" è un gioco che associa ai classici strumenti del gioco di società tutte le caratteristiche del video-game, in cui dati, eventi e messaggi vengono gradevolmente mostrati sul video, all'interno di una grande finestra posta al centro del tabellone di gioco. A testimoniare la serietà del gioco vi è la grande mole del listato, capace di garantire la varietà di situazioni richieste da chi ama il gioco da tavolo. Il gioco è riservato a due giocatori, uno dei quali può essere all'occorrenza sostituito dallo Spectrum (a patto che tiriate i dadi e prendiate decisioni in sua vece). Sco-

po del gioco è accumulare il maggior numero di punti possibili, in attesa che l'avversario esaurisca i suoi soldi. Tene presente che ad ogni vostro movimento sul tabellone, o per raggiungere una delle mete turistiche verso cui potete dirigervi, corrisponde il consumo di un po' di benzina: esaurendo il carburante potrete usufruire di un traino (costoso!) fino a quando raggiungerete una stazione di rifornimento. Comprate dunque benzina quando potete, visto anche che al termine della partita tutto ciò che si trova nel serbatoio viene trasformato in punti. Insomma, le tre variabili da considerare (punti, soldi, benzina) sono in qualche modo interrelate, e tale circostanza richiede che il giocatore elabori una idonea strategia di gioco.

Per quanto concerne il listato, possiamo distinguere quattro parti fondamentali: 1) la inizializzazione, il set grafico, la presentazione e le istruzioni, l'allestimento del tabellone di gioco; 2) la routine "tiro del dado" per stabilire i vostri movimenti; 3) il movimento dei segnalini dei giocatori, ed eventualmente la chiamata di una subroutine che attiva situazioni particolari;

4) la routine di fine gioco.

Lanciando il programma, viene eseguita la prima parte, quindi le linee 120 e 130 fanno girare continuamente la seconda e la terza parte, sino a quando si verifica la condizione di fine gioco (esaurimento dei soldi), giungendo così alla quarta parte.

Le quattro sezioni operano come segue: Parte 1: le linee sino alla 110 chiamano tutte le routine di inizializzazione e allestimento gioco. In particolare la routine 9000 mostra le istruzioni e carica il set grafico, mentre la routine 8000 si occupa della realizzazione del tabellone di gioco. Quest'ultimo viene realizzato utilizzando sequenze appropriate di PLOT/DRAW e lasciando libero il centro dello schermo per le varie routine grafiche che saranno richiamate nel corso della partita. Le singole caselle sono numerate combinando variamente INK, PAPER e BRIGHT, combinazioni controllate dai DATA posti alla linea 8000. Sono proprio gli attributi di ogni casella a decidere il destino del giocatore che atterra su di essa, ed in tal senso è necessario porre particolare cura nella fase di eventuale battitura del listato: alterando gli attributi non viene infatti chiamata la subroutine corretta.

Parte 2: la routine che si occupa del tiro dei dadi è abbastanza semplice. La realizzazione grafica del dado avviene fra le linee 8300-8420, mentre l'output viene realizzato chiamando la subroutine 8430. Per lanciare il dado è sufficiente premere ENTER e mantenerlo premuto sino a quando non credete di aver realizzato il punteggio desiderato. Il rilasciamento del tasto fa arrestare il dado. Parte 3: per realizzare il corretto movimento dei pezzi sul tabellone il programma si avvale di una serie di state-

Descrizione del programma

Linee

50- 110

120- 130

140- 230

1000-1090

2090-2260

3940-4160

4340-4490

4495-4720

4740-4790

4795-5070

5150-5590

5600-5900

5990-6350

6490-6660

7990-8270

8300-8420

8425-8490

8495-8500

8505-8530

8590-8660

9000-9050

9100-9140

9900-9920

Commento

inizializzazione e chiamata subroutine istruzioni, UDG, allestimento tabellone, ecc.

verifica consistenza dei capitali, cambio turno, chiamata subroutine per tiro del dado, chiamata routine movimento

routine di fine gioco

verifica carburante

ristorante

castello

stazione di rifornimento

posto per campeggio e negozio

tintarella

golf club

mare e slot machine

ferrovia

zoo safari

routine per il movimento

allestimento tabellone di gioco

realizzazione del dado

tiro del dado

pulisce il centro dello schermo

pulisce il margine destro dello schermo

riepilogo punteggi

istruzioni

set grafico

salvataggio e verifica programma

Descrizione delle variabili

C(1), C(2)

capitale dei giocatori

J(1), J(2)

punteggi

P(1), P(2)

benzina

H(1), H(2)

coordinata orizzontale dei giocatori

V(1), V(2)

coordinata verticale dei giocatori

PL

giocatore

D

valore dei punti realizzati lanciando il dado

LO

indica il cambio del turno di gioco

MI

miglia da percorrere per raggiungere una località

TI

esame del carburante

SUB

subroutine attivata

Z\$

INPUT LINE

Altre variabili e matrici vengono utilizzate nell'ambito di ciascuna subroutine: il loro confinamento all'interno di tali cicli ne consente la facile identificazione.

ment FOR... NEXT. Ogni loop verifica che il movimento non conduca il segnalino oltre i margini del tabellone: in caso contrario si passa al loop successivo. Tale controllo poteva essere realizzato anche attraverso i classici IF... THEN, ma con questi ultimi il movimento sarebbe stato molto più lento.

Giungendo su una casella ne vengono letti gli attributi. Se la combinazione di inchiostro, carta e luminosità ritorna il valore 104, il turno passa all'avversario (tramite la linea 6600). Per ogni valore diverso da 104 l'attributo viene invece tradotto in un numero di linea e la appropriata subroutine è chiamata con uno statement GO SUB. Nell'ambito di ogni subroutine (sulle quali manteniamo il più stretto riserbo) possono capitare eventi felici, altri sgradevoli, altri catastrofici: è chiaro che in questo modo potrete guadagnare o perdere punti, soldi e benzina. Ricordatevi che se rima-

nete "a secco", dovrete pagare il traino, e se questo avviene durante il movimento verso una meta turistica, perderete il resto del turno. Il gioco termina quando un giocatore esaurisce i soldi, ma il vincitore è chi ha accumulato il maggior numero di punti: verso la fase finale, quindi, la strategia diventa particolarmente importante (ad esempio, chi è in notevole vantaggio di punti può tentare di esaurire velocemente i suoi quattrini, in modo da concludere la partita senza aver dato tempo all'avversario di rifarsi).

Parte 4: quando uno dei giocatori ha esaurito il capitale, la linea 130 cessa di sostenere il loop principale, e viene dunque attivata la routine di fine gioco. Viene a questo punto mostrato il riepilogo finale, tramite la chiamata della subroutine 8600 (spesso richiamata anche nell'ambito della partita).

Per battere il listato

Con listati di tali dimensioni, commettere un errore è quasi inevitabile. Per ovviare a tale inconveniente vi consigliamo di batterlo in sezioni. Iniziate a battere dall'inizio sino alla linea 230, saltate tutta la parte 1000-6350 e sostituite la temporaneamente con la linea 6480 RETURN, quindi inserite l'ultima parte, dalla linea 6490 sino alla fine. A questo punto, avviando il programma, potrete verificare la correttezza della battitura (inizializzazione, grafica, istruzioni, movimento, dadi...), mentre mancano ovviamente tutte le subroutine. Queste ultime potranno essere inserite una alla volta, controllandone l'efficienza. Per controllare ogni subroutine non è necessario aggirarsi sul tabellone in attesa di capitare fortuitamente sulla casella giu-

sta: fate invece stampare il tabellone di gioco, quindi arrestate il programma ed inserite come comando diretto GO TO (linea desiderata). Il numero di linea è ovviamente quello della prima linea della subroutine (può essere quello della REM).

Poiché il programma prevede un INPUT LINE Z\$, che disabilita BREAK e STOP, l'arresto del programma stesso può essere realizzato solo premendo simultaneamente CAPS SHIFT ed il tasto 6, che realizza lo "stop in input" nel caso del LINE (strano, no?). La subroutine di verifica carburante (linee 1000-1090) è di difficile verifica: inseritela dunque per ultima, quando siete certi del funzionamento di tutte le altre. Tenete infine presente che gli UDG veramente vitali per il corretto

funzionamento del programma sono quelli associati ai segnalini dei giocatori (A & B) e ai punti del dado (C).

Una volta salvato il programma (con GOTO 9900) fatelo girare: dapprima vedrete le istruzioni, quindi verrà allestito il tabellone. Il computer deciderà in modo del tutto casuale chi deve giocare per primo. Il segnalino del giocatore a cui tocca la mossa successiva è mostrato all'estremo margine destro dello schermo. Se il giocatore arriva ad una casella associata ad una subroutine, segua le istruzioni di cui ogni subroutine stessa è dotata. Per entrare nello spirito del gioco, infine, immaginate di essere a spasso con la vostra macchina e di avere così l'occasione di visitare posti interessanti, quali vecchi castelli, zoo safari, il mare e così via. Buone vacanze.

```
10 REM *****
20 REM *   GITA  TURISTICA  *
30 REM *****
40 REM
50 CLS : PRINT AT 11,9; PAPER 5;"FERMA IL NASTRO": GO SUB 8650
60 RANDOMIZE : PAPER 7: INK 0:
  CLS : GO SUB 9000: LET PL=1+
  INT (RND*2): BORDER PL: CLS :
  PRINT AT 12,6;"GIOCATORE ";
  INK PL;CHR$(143+PL); INK 0;" INIZIA": BEEP .7,20: GO SUB 8650:
  CLS : BORDER 0
70 DIM V(2): DIM H(2): DIM J(2):
  DIM C(2): DIM P(2)
80 FOR N=1 TO 2: LET J(N)=0: LET C(N)=250: LET P(N)=5: NEXT N
90 LET FWD=2: LET FSF=0: LET TI=0:
  LET D=1
100 LET V(1)=1: LET V(2)=1: LET H(1)=1: LET H(2)=2
110 GO SUB 8000
120 LET LO=0: GO SUB 8430: GO SUB 6500
130 IF C(1)>0 AND C(2)>0 THEN
  GO TO 120
140 GO SUB 8500: PRINT AT 7,4; INK 1;
  FLASH 1; BRIGHT 1;" LA PARTITA E' FINITA ";AT 10,4; PAPER 5;
  FLASH 0;"GIOCATORE "; INK 1+(C(2)<=0);CHR$(144+(C(2)<=0));
  INK 1;" SENZA SOLDI";AT 14,4;"P REMI ENTER PER I PUNTI": INPUT
  LINE Z$
150 LET J(1)=J(1)+P(1): LET J(2)=J(2)+P(2): LET P(1)=0: LET P(2)=0:
  GO SUB 8600
160 IF J(1)=J(2) THEN GO SUB 8500:
  PRINT AT 7,4;" PARTITA PATTATA ": GO SUB 8650: GO TO 190
170 GO SUB 8500: PRINT AT 7,4;" GIOCATORE "; INK 1+(J(2)>J(1));
  CHR$(144+(J(2)>J(1))); INK 0;" HA VINTO!"
180 IF INKEY$<>" " THEN GO TO 180
```

```
190 PRINT AT 9,4;" UN'ALTRA PARTITA ? ";AT 12,4;" PREMI ";
  INVERSE 1;"S"; INVERSE 0;"I O "; INVERSE 1;"N"; INVERSE 0;"O ";AT 14,4;" OPPURE ";
  INVERSE 1;"D"; INVERSE 0;"IVERS O, PER ";AT 15,4;"CARICARE UN ALTRO GIOCO"
195 IF INKEY$="S" OR INKEY$="s" THEN RUN 60
200 IF INKEY$="N" OR INKEY$="n" THEN GO SUB 8500: GO TO 230
210 IF INKEY$="D" OR INKEY$="d" THEN CLS : PRINT AT 10,5;
  PAPER 5;"FAI PARTIRE IL NASTRO": LOAD ""
220 GO TO 190
230 FOR N=1 TO 13: PRINT AT 10,4;"GRAZIE PER LA PARTITA..";AT 11,10;"CIAOOOO...": BEEP .2,2*N:
  NEXT N: STOP
1000 REM ## VERIFICA PETROLIO ##
1010 LET TI=TI+1: LET P(PL)=P(PL)-(INT(MI/.3))/100: IF P(PL)>0 THEN RETURN
1020 LET DTV=D: LET LO=1: LET P(PL)=0: GO SUB 8500: CIRCLE 130,90,32:
  CIRCLE 130,90,30: PLOT 115,80: DRAW 30,0,-5*PI: PLOT 117,82:
  DRAW -5,20: PRINT AT 12,14;"fue 1";AT 9,14; INK 2;"{SG8} ";
  AT 9,16; INK 1;"{G5} {SG8}"
1030 PRINT AT 4,4;" HAI FINITO LA BENZINA "
1040 PRINT AT 16,4;"DEVI PAGARE ~ 10 + TIRO";AT 17,4;"DEL DADO PER RIMORCHIO": FOR N=1 TO 80: NEXT N:
  PAUSE 50: GO SUB 8500: GO SUB 8430: LET C(PL)=C(PL)-(10+D)
1050 LET J(PL)=J(PL)-5*(TI=1)-2*(TI=2): LET P(PL)=.75: GO SUB 8610
1060 GO SUB 8650: GO SUB 8500: IF TI=1 THEN PRINT AT 10,4;"GIORNO SFORTUNATO ! ";AT 11,4;"PERDI 5 PU
```

```

NTI !";AT 12,4;"PUNTEGGIO ATTUAL
E: ";J(PL)
1070 PRINT AT 16,4;"HAI PAGATO ~ ";1
0+D;" ";AT 17,4;"CASSA ATTUALE
: ";C(PL);" ": IF TI=2 THEN
PRINT AT 10,4;"PERDI 2 PUNTI !
";AT 11,4;"PUNTEGGIO ATTUALE: ";
J(PL)
1080 GO SUB 8650
1090 GO SUB 8500: GO SUB 8430: LET D=
DTV: RETURN
2090 REM #### RISTORANTE ####
2100 GO SUB 8650: GO SUB 8500:
PRINT AT 5,5;"FERMATI UN MOMENT
O";AT 6,5;"PER MANGIARE ";
AT 7,5;"QUALCOSA PRESSO ";AT 8
,5;"UNA TRATTORIA ":
GO SUB 8650
2110 GO SUB 8500: PLOT 55,138: DRAW 8
1,0: DRAW 0,-100: DRAW -81,0:
DRAW 0,100
2120 LET SCM=0: DIM T$(8,10)
2130 DATA "OSTRICHE","GAMBERI","MELON
E","MINESTRA"
2140 DATA "PESCE","POLLO","BISTECCA",
"PIZZA"
2150 DATA "GELATO","GATEAUX","CROSTAT
A","FRITTELLA"
2160 DATA "SPREMUTA","THE","CAFFE',"",
"COCACOLA"
2170 FOR M=10 TO 40 STEP 10: LET NN=1
+INT (RND*4): RESTORE (2120+M):
FOR N=1 TO NN: READ T$(M/10):
NEXT N: LET SCM=SCM+NN: NEXT M
2180 FOR N=5 TO 16: PRINT AT N,7;T$(5
): NEXT N: PRINT AT 5,9; PAPER 3
; INK 7; BRIGHT 1;" MENU "
2190 FOR N=7 TO 13 STEP 2: PAUSE 25:
BEEP .02,N: PRINT AT N,7;T$( (N-
5)/2): NEXT N
2200 LET T$(6)="YUK !!": LET T$(7)="N
ON MALE": LET T$(8)="YUMMY"
2210 PAUSE 70: PRINT AT 12,17;
PAPER 5;"HAI SCELTO";AT 13,17;T
$( (6+SCM>8)+(SCM>11)): BEEP .4,1
-20*(SCM<9)+20*(SCM>11): PRINT
AT 14,17;"PUNTI ";INT (SCM/2):
LET J(PL)=J(PL)+INT (SCM/2)
2220 GO SUB 8650: PRINT AT 12,17;
PAPER 4;" PREMI ";AT 13,17;"
ENTER ";AT 14,17;"PER AVERE
";AT 15,17;"IL CONTO ";AT 16,17
;" ~1+DADO "
2230 IF INKEY$<>" " THEN GO TO 2230
2240 GO SUB 8650: GO SUB 8500:
GO SUB 8430: LET C(PL)=C(PL)-1-
D: PRINT AT 14,6;" IL CONTO E'
~";1+D
2250 IF SCM>11 AND (1+D)<3 THEN
PRINT AT 16,6;"BUONO & ECONOMIC
O";AT 17,6;"BONUS --> PUNTI 2":
LET J(PL)=J(PL)+2: FOR N=1 TO 1
00: NEXT N
2260 RETURN
3940 REM #### CASTELLO ####

```

```


3950 GO SUB 3960: GO SUB 3990:
RETURN
3960 PRINT AT 5,5; PAPER 6;"{SG8}
{G8}{SG8}{G8}{SG8}
{SG8}{G8}{SG8}{G8}{SG8}";AT 6,5;
"{5SG8} {5SG8}";AT 7,
5;"{G5}{3SG8}{SG5}
{G5}{3SG8}{SG5}"
3970 FOR N=1 TO 8: PRINT AT N+7,5;
PAPER 6;"{G5}{20SG8}{SG5}":
NEXT N: FOR N=11 TO 15: PRINT
AT N,14; PAPER 7;"{5SG2}":
NEXT N
3980 PRINT AT 9,6; PAPER 2; INK 7;
BRIGHT 1;"CASTELLO MISTERIOSO"
: RETURN
3990 LET MI=15+INT (RND*15): PRINT
AT 17,4; PAPER 5;"UN VIAGGIO DI
";MI;" MIGLIA";AT 14,20; PAPER 4
;"ENTRATA";AT 15,20;"~ 1.50 "
4000 INPUT "VUOI ANDARE A VISITARLO (
S/N)?"; LINE Z$: IF Z$="n" OR Z$
="N" THEN RETURN
4010 GO SUB 1000: IF LO=1 THEN LET T
I=0: RETURN
4015 IF 1+INT (RND*14)=1 THEN
PAUSE 50: PRINT AT 14,20;" OGGI
";AT 15,20;"CHIUSO ": LET J(PL
)=J(PL)-5: PRINT AT 16,4;
PAPER 4;"DISAPPUNTO!!!";AT 17,4
;"PERDI 5 PUNTI ";
FOR N=10 TO 1 STEP -1: BEEP .3,
N: NEXT N: GO SUB 8650: GO SUB 1
000: LET TI=0: GO TO 8600:
RETURN
4020 LET C(PL)=C(PL)-1.5: RESTORE 403
0: DIM H$(6,12)
4030 DATA "NOIOSO!","BRUTTINO","COSI'
COSI'","AFFASCINANTE","ECCITANT
E","FAVOLOSO"
4040 FOR N=1 TO 6: BEEP .1,N: READ Q$
: LET H$(N)=Q$: NEXT N: PRINT
AT 4,4; PAPER 5;"TIRA IL DADO PE
R VEDERE";AT 5,10;"SE E' BELLO"
4050 GO SUB 8650: FOR N=1 TO 200:
NEXT N: IF INKEY$<>" " THEN
GO TO 4050
4060 GO SUB 8500: GO SUB 8430: FOR N=
1 TO 300: NEXT N: GO SUB 3960:
PRINT AT 4,9; PAPER 6;"E' ";H$(
D);AT 5,13;"PUNTI ";D: LET J(PL)
=J(PL)+D: LET FC=1: GO TO 4080
4070 GO SUB 8650: PRINT AT 4,4;
PAPER 6;"
";AT 5,10;" ":
RETURN
4080 IF RND<.2 THEN GO SUB 4070:
PRINT AT 4,4;"HAI VISTO IL TERR
IBILE": PAUSE 50: PRINT AT 5,10;
FLASH 1; PAPER 3;" FANTASMA "
; FLASH 0; PAPER 6;AT 7,10;" PU
NTI 5 ": GO SUB 4130: LET J(PL)
)=J(PL)+5: LET FC=0


```

```

4090 IF RND<.3 THEN GO SUB 4070:
      PRINT AT 4,4;"BELLE FOTO DALLA
TORRE";AT 7,10;" PUNTI 3 ":
      LET J(PL)=J(PL)+3: LET FC=0
4095 IF RND<.25 THEN GO SUB 4070:
      PRINT AT 4,4; PAPER 3; INK 7;
      BRIGHT 1;"TI SEI PERSO NEI ";
      AT 5,10;"SOTTERRANEI ";AT 7,10;"
PERDI 4 PUNTI": LET J(PL)=J(PL)-
4: LET FC=0
4100 IF RND<.3 THEN GO SUB 4070:
      PRINT AT 4,4; PAPER 4;"TI VIENE
OFFERTO UN";AT 5,10;"OTTIMO THE
' ";AT 7,10;" PUNTI 4 ":
      LET J(PL)=J(PL)+4: LET FC=0
4110 IF FC THEN GO TO 4080
4120 GO SUB 8650: GO SUB 8500:
      GO SUB 1000: LET TI=0: GO SUB 8
600: RETURN
4130 FOR N=1 TO 10
4140 PRINT AT 10,20; OVER 1; PAPER 7;
      INK 0;"E"
4145 PRINT AT 11,20; OVER 1; PAPER 7;
      INK 0;"HI"
4150 PRINT AT 12,20; OVER 1; PAPER 7;
      INK 0;"LL"
4160 PAUSE 20: NEXT N: RETURN
4340 REM #STAZIONE DI SERVIZIO#
4350 IF P(PL)>=7 OR RND<.4 OR LO
      THEN GO TO 4497
4360 GO SUB 8610: IF P(PL)<=0 AND C(P
L)=0 THEN GO TO 140
4365 GO SUB 8500: PRINT AT 5,4;"PUOI
COMPRARE BENZINA";AT 6,4;"ORA HA
I ";P(PL);" GALLONI";AT 8,4;"SER
BATOIO: MAX 10 GAL";AT 14,12;"CA
SSA ~ ";C(PL);" "
4370 DIM H$(5,6)
4380 LET H$(1)="{5SG8}"
4390 LET H$(2)="{SG8} {SG8}"
4400 LET H$(3)="{SG8} {2SG8}"
4410 LET H$(4)="{5SG8}{G5}"
4420 LET H$(5)="{5SG8}{SG2}"
4430 DATA 1,2,3,4,4,4,5,1
4440 RESTORE 4430: FOR N=1 TO 8:
      READ A: PRINT AT 8+N,5;H$(A):
      NEXT N: LET PR=160+INT (RND*10)
: PRINT AT 15,6;PR
4445 INPUT "QUANTI GALLONI COMPRI? ";
      LINE R$: IF CODE R$<48 OR
CODE R$>58 THEN GO TO 4445
4450 LET GA=VAL R$: IF GA*(PR/100)>C(
PL) THEN LET GA=C(PL)/(PR/100):
      LET P(PL)=P(PL)+GA: LET C(PL)=0
: GO SUB 8610: PRINT AT 9,12;"HA
I SOLDI SOLO";AT 10,12;"PER GALL
ONI ";(INT GA*100)/100;AT 14,20
;"0 ": GO TO 4470
4454 IF P(PL)+GA>10 THEN GO TO 4445
4455 IF P(PL)+GA<=10 THEN LET P(PL)=
P(PL)+GA: LET C(PL)=C(PL)-GA*(PR
/100)
4465 GO SUB 8610: PRINT ;AT 10,11;"BE
NZINA ";P(PL);" GAL";AT 14,12;"C
ASSA ~ ";C(PL);AT 6,4; PAPER 6;"

```





```

4470 GO SUB 8650: GO SUB 8500
4490 IF LO OR RND>.5 THEN RETURN
4495 REM ## CAMPING & SPESA ##
4497 IF FSF=1 THEN GO TO 4550
4500 GO SUB 8500: FOR N=5 TO 7:
      PRINT AT N,4;"
      ": NEXT N: PRINT AT 6,4;
      PAPER 4;" UN POSTO PER CAMPING?
"
4510 LET SF=3+INT (RND*3): PRINT AT 9
,4;"DEVI PAGARE LA TASSA DI";
      AT 10,4;"SOGGIORNO. PREMI ENTER
";AT 11,4;"PER CONOSCERE L'IMPOR
TO";AT 14,4;"IL COSTO SARA' ~ ";
      SF;" PER";AT 15,4;"OGNI PUNTO DE
L DADO"
4515 PAUSE 0: BEEP .2,20: FOR N=1
      TO 150: NEXT N: IF INKEY$<>"
      THEN GO TO 4500
4520 GO SUB 8500: GO SUB 8430:
      PRINT AT 11,4;"IL COSTO E' ~ ";
      SF*D;" ";: LET C(PL)=C(PL)- (
      SF*D)
4530 DIM N$(3,9): LET N$(1)="TERRIBIL
E": LET N$(2)="CARINO": LET N$(3
)="BELLO!": LET ST=1+INT (RND*3)
4540 PAUSE 100: LET J(PL)=J(PL)+(ST=3
)+(ST=2 OR ST=3)-(ST=1): PRINT
      AT 15,4;"E' UN POSTO ";N$(ST);
      AT 16,4;"PUNTI ";(ST=3)+(ST=2
      OR ST=3)-(ST=1): LET FSF=2:
      GO SUB 8650
4545 IF RND<.4 THEN GO TO 8600
4550 GO SUB 4555: GO TO 4590
4555 GO SUB 8500: FOR N=8 TO 11:
      PRINT AT N,5; INK 3;"{21SG8}":
      NEXT N: FOR N=9 TO 10: PRINT
      AT N,7; INK 1;"{SG6}{G6}{SG6}
{G6}{SG6}{G6}{SG6}{G6}{SG6}{G6}
{SG6}{G6}{SG6}{G6}{SG6}{G6}
{SG6}": NEXT N
4560 PRINT AT 10,7; PAPER 7; INK 2;
      BRIGHT 1;"GRANDI MAGAZZINI"
4570 FOR N=12 TO 17: PRINT AT N,5;
      INK 3;"{2SG8} {SG8}
{2SG8}": NEXT N
4580 PRINT AT 17,5; INK 1; PAPER 3;"
{SG8}{6G7}{7SG4}{SG8}": RETURN
4590 PRINT AT 5,5;"DEVI FAR PROVVISTE
": GO SUB 8650
4600 PRINT AT 6,5; PAPER 4;"IL DADO D
ECIDERA' SE";AT 7,5;"IL NEGOZIO
E' BUONO": GO SUB 8650:
      GO SUB 8500: GO SUB 8430
4610 RESTORE 4610: DATA "INDECENTI!",
"CARI","NORMALI","ECONOMICI","RI
SPARMIOSI","BASSISSIMI"

```

```

4620 FOR N=1 TO D: READ QS: NEXT N:
      LET DV=D: PRINT AT 12,4;"I PREZ
      ZI NEL NEGOZIO";AT 13,4;"SONO ";
      QS
4630 PRINT AT 15,5;"PUNTI ";D: LET J(
      PL)=J(PL)+D: GO SUB 8650
4640 GO SUB 8500: LET PR=12+INT (RND*
      5): PRINT AT 12,4;"PENSAVI DI SP
      ENDERE ~";PR;AT 13,4;"IL DADO TI
      DIRA' QUANTO";AT 14,4;"HAI REAL
      MENTE SPESO": PAUSE 90: GO SUB 8
      430
4650 LET PP=5+D*3: PRINT AT 13,4;"HAI
      SPESO ~ ";PP;"          "+" "
      AND PP<10);AT 14,4; PAPER 6;"
      "
4660 LET C(PL)=C(PL)-PP
4670 IF PP<PR-2 THEN PRINT AT 15,4;"
      HAI RISPARMIATO": IF DV=1 OR DV=
      2 THEN PRINT AT 15,19;" PUR CON
      ";AT 16,4;"PREZZI ";QS;" "
4680 IF PP<PR-2 THEN PRINT AT 17,18;
      "PUNTI ";3+3*(DV<3): LET J(PL)=J
      (PL)+3+3*(DV<3): GO TO 4720
4690 IF PP>PR+2 THEN PRINT AT 15,4;"
      HAI SPESO TANTO": IF DV>4 THEN
      PRINT AT 15,19;" PUR CON";AT 16
      ,4;"PREZZI ";QS;" "
4700 IF PP>PR+2 THEN PRINT AT 17,18;
      "PUNTI ";-3-3*(DV>4): LET J(PL)=
      J(PL)-3-3*(DV>4): GO TO 4720
4710 PRINT AT 16,4;"HAI RISPETTATO IL
      TUO";AT 17,4;"BUDGET. PUNTI 2"
      : LET J(PL)=J(PL)+2
4720 LET FSF=0: GO SUB 8650: GO TO 86
      00
4740 REM ### TINTARELLA ###
4750 GO SUB 8500: LET SCD=0: PLOT 39,
      120: DRAW 50,10,-.5*PI: FOR N=1
      TO 5: DRAW -10,-2,.5*PI:
      NEXT N: PLOT 64,125: DRAW 5,-40
      : PLOT 31,64: DRAW 183,0: PLOT 1
      00,104: DRAW 30,30: DRAW 30,-30:
      PLOT 130,134: DRAW 50,8: DRAW 3
      0,-30: DRAW -50,-8: DRAW -56,0
4760 CIRCLE 90,80,7: DRAW 30,0:
      DRAW 10,16: DRAW 14,-16: DRAW 5
      ,5: PLOT 100,77: DRAW 22,-4:
      PRINT AT 14,4;"UN GIORNO DI RIP
      OSO. ": LET PUL=1
4770 PRINT AT 15,4;"PRENDERAI LA TINT
      ARELLA";AT 16,4;"FINCHE' NON FAR
      AI 3 O ";AT 17,4;"MENO DI 3. PR
      EMI ENTER.": PAUSE 0: FOR N=6
      TO 8: PRINT AT N,6; PAPER 6;"(
      " AND PUL=1): NEXT N: LET PUL=
      0: GO SUB 8430: PRINT AT 10,20;
      FLASH 1;"ANCORA": LET J(PL)=J(P
      L)+D: LET SCD=SCD+D: IF D>3
      THEN GO TO 4770
4780 PRINT AT 10,4; PAPER 5;"IL SOLE
      SE N'E' ANDATO!": FOR N=0 TO -10
      STEP -1: BEEP .05,N: NEXT N:
      IF SCD<=8 THEN GO SUB 2000
4790 GO SUB 8650: GO SUB 8500:
      GO TO 8600

```

```

4795 REM ### GOLF ###
4800 GO SUB 4810: GO SUB 4860:
      RETURN
4810 FOR N=4 TO 9: PRINT AT N,4;
      PAPER 5;"
      ": NEXT N
4820 FOR N=10 TO 17: PRINT AT N,4;
      PAPER 4;"
      ": NEXT N
4830 PRINT AT 12,6; PAPER 4; INK 6;"D
      EG";AT 13,6;"H{SG8}I";AT 14,6;"J
      KM"
4850 PLOT 180,120: DRAW 0,-24:
      PRINT AT 7,23;"1": RETURN
4860 PRINT AT 6,6;"UN GIORNO AL";AT 7
      ,6;"'GOLF CLUB'": PAUSE 75:
      PRINT AT 12,12;"SFIDA IL TUO";
      AT 13,12;"AVVERSARIO IN";AT 14,1
      2;"QUESTA PROVA."
4870 GO SUB 8650: GO SUB 4810:
      PRINT AT 6,6;"CHI PERDE PAGA ";
      AT 7,6;"IL CAMPO (~10).";AT 8,6;
      "IL VINCITORE ";AT 9,6;"PRENDE
      10 PUNTI"
4880 DIM S(2,18)
4890 GO SUB 8650: GO SUB 8500:
      PRINT AT 10,4;"GIOCATORE ";
      INK 1;"A"; INK 0;" GIOCATORE ";
      INK 2;"B";AT 4,4;"PREMI <V> PER
      VELOCITA'"
4900 DATA 2,3,4,3,4,2,3,3,4,4,2,3,5,4
      ,3,2,4,3
4910 LET DR=0: LET GSA=0: LET GSB=0:
      RESTORE 4900: LET HOL=1+INT (
      RND*80): FOR N=1 TO 18
4920 READ P: PRINT AT 6,6;"BUCA NUMER
      O ";N;AT 7,6;"NORMA = ";P+1
4930 LET S(1,N)=P+INT (RND*3)
4940 LET S(2,N)=P+INT (RND*3)
4950 LET GSA=GSA+S(1,N): LET GSB=GSB+
      S(2,N)
4960 PRINT AT 12,4;"TIRO";AT 15,4;"TO
      T.";AT 12,11;S(1,N);AT 12,23;S(2
      ,N);AT 15,11;GSA;AT 15,23;GSB
4970 IF GSA=HOL AND DR=0 THEN
      PRINT AT 12,11;"1": FOR M=1
      TO 40 STEP 4: BEEP .4,M:
      PRINT AT 17,4; FLASH 1;"BUCA IN
      UN COLPO-->"; INK 1;"A": NEXT M
      : LET GSA=GSA-S(1,N)+1: PRINT
      AT 15,11;GSA: LET DR=1: PRINT
      AT 17,4; PAPER 6;"
      "
4980 IF GSB=HOL AND DR=0 THEN
      PRINT AT 12,23;"1": FOR M=1
      TO 40 STEP 4: BEEP .4,M:
      PRINT AT 17,4; FLASH 1;"BUCA IN
      UN COLPO-->"; INK 2;"B": NEXT M
      : LET GSB=GSB-S(2,N)+1: PRINT
      AT 15,23;GSB: LET DR=2: PRINT
      AT 17,4; PAPER 6;"
      "
4990 PAUSE 150: BEEP .5,20: NEXT N:
      PAUSE 50: GO SUB 4810
4995 IF GSA=GSB THEN PAUSE 100:
      PRINT AT 5,4;"PARTITA PATTA";

```

```

AT 6,4;"DIVIDETE COSTO E PUNTI":
LET J(1)=J(1)+5: LET J(2)=J(2)+
5: LET C(1)=C(1)-5: LET C(2)=C(2)
)-5
5000 IF GSA>GSB THEN LET J(2)=J(2)+1
0: LET C(1)=C(1)-10: PRINT AT 5,
4;"GIOCATORE "; INK 2;"B"; INK 0
;" HA VINTO!"
5010 IF GSA<GSB THEN LET J(1)=J(1)+1
0: LET C(2)=C(2)-10: PRINT AT 5,
4;"GIOCATORE "; INK 1;"A"; INK 0
;" HA VINTO!"
5015 IF DR=0 THEN GO TO 5060
5020 IF DR=1 THEN PRINT AT 7,4;"GIOC
ATORE "; INK 1;"A"; INK 0;" HA";
AT 8,4;"FATTO UN'A BUCA";AT 9,4;"
IN UNO": LET U$="A"
5030 IF DR=2 THEN PRINT AT 7,4;"GIOC
ATORE "; INK 2;"B"; INK 0;" HA";
AT 8,4;"FATTO UN'A BUCA";AT 9,4;"
IN UNO": LET U$="B"
5040 GO SUB 8650: GO SUB 4810: IF DR>
0 THEN PRINT AT 11,10;"UN COLPO
ALLA";AT 12,10;"19' BUCA COSTA"
;AT 13,10;"~2 PER OGNI";AT 14,10
;"PUNTO DEL DADO";AT 15,10;"SOLO
PER "; INK DR;U$: PAUSE 200:
GO SUB 8430
5050 IF DR>0 THEN LET C(DR)=C(DR)-2*
D: PRINT AT 17,5;"COSTO ~";2*D
5060 GO SUB 8650
5070 GO TO 8600
5150 REM # SPIAGGIA & SLOT MAC #
5160 LET D=6: GO SUB 5170: GO TO 5230
5170 LET SUB=5170: FOR N=4 TO 10:
PRINT AT N,4; PAPER 5;"
": NEXT N: FOR N=
11 TO 12: PRINT AT N,4; PAPER 1;
"
":
NEXT N
5180 IF D>=4 THEN FOR N=1 TO 10:
INK 6: PAPER 5: CIRCLE 160,120,
N: NEXT N: INK 0: PAPER 7
5190 IF D<=3 THEN PRINT AT 6,18;
PAPER 5; INK 7;"DEFEFEGF";AT 7,
18;"H{6SG8}I";AT 8,18;"H{6SG8}I"
;AT 9,18;"JKLKLKLM"
5200 IF D<=2 THEN FOR N=10 TO 16
STEP 2: FOR M=11 TO 17 STEP 2:
PRINT AT N,18;"' ' ' ' ";AT M,
16;" ' ' ' ' ": NEXT M: NEXT N
5210 IF D=1 THEN PLOT 180,90: INK 5:
BRIGHT 1: DRAW -5,-10: DRAW 3,0
: DRAW -5,-15: INK 0: BRIGHT 0
5220 RETURN
5230 FOR N=13 TO 17: PRINT AT N,13;
PAPER 7; INK 3;"{SG5}
{G5}": NEXT N: PRINT AT 9,4;"GIT
A ALLA BAI A SABBIOSA";AT 13,13;
INK 3;"{SG4}{10G3}{G7}";AT 17,1
3;"{SG1}{10SG3}{SG2}";AT 14,14;
INK 1;"PARCHEGGIO";AT 15,18;"~1
";AT 16,15;"AL GIORNO": GO SUB 8
650

```

LOAD

Gita turistica

```

5235 FOR N=13 TO 17: PRINT AT N,13;
PAPER 6;"
": NEXT N
5240 LET MI=20+INT (RND*10): PRINT
AT 13,5;"VUOI ANDARE A VEDERE";
AT 14,5;"IL MARE, DISTANTE";AT 1
5,5;MI;" MIGLIA ?": INPUT "RISPO
NDI "; FLASH 1;"S"; FLASH 0;" O
"; FLASH 1;"N"; FLASH 0;" ";
LINE Z$: IF Z$="N" THEN
RETURN
5245 GO SUB 1000: IF LO=1 THEN LET T
I=0: RETURN
5250 LET C(PL)=C(PL)-1: LET LO=0:
GO SUB 8500: PRINT AT 11,4;"TIR
A IL DADO PER VEDERE";AT 12,4;"C
HE TEMPO FA. FARAI ";AT 13,4;
"PIU' PUNTI SE IL ";AT 14,
4;"TEMPO SARA' BELLO. "
5260 GO SUB 8430: DIM H$(6,7)
5270 LET H$(1)="BRUTTO!": LET H$(2)="
PIOVOSO": LET H$(3)="COPERTO":
LET H$(4)="CALDO": LET H$(5)="B
ELLO!": LET H$(6)="SUPER!!"
5280 PAUSE 25: GO SUB 8500: GO SUB SU
B: LET J(PL)=J(PL)+D: PRINT AT 1
0,4;"IL TEMPO E' ";: PAUSE 50:
PRINT AT 11,4;H$(D);AT 12,4;"PU
NTI ";D
5290 IF SUB<>5170 THEN RETURN
5300 IF D>2 AND RND<.6 THEN GO SUB 8
650: PRINT AT 5,4;"HAI VINTO UNA
";AT 6,4;"GARA DI";AT 7,4;"CASTE
LLI DI SABBIA": LET PRI=2+INT (
RND*4): PRINT AT 8,4;"PUNTI ";PR
I: LET J(PL)=J(PL)+PRI
5305 IF D>2 THEN GO TO 5580
5310 PAUSE 70: PRINT AT 5,4;"ENTRI IN
UN";AT 6,4;"PALAZZO PER";AT 7,4
;"RIPARARTI DALLA";AT 8,4;"PIOGG
IA: E' UN...": GO SUB 8650
5320 GO SUB 8500: FOR N=5 TO 17
STEP 2: PRINT AT N,4+INT N/2;
PAPER 1+INT (RND*5); INK 9;
FLASH INT (RND*2); BRIGHT INT (
RND*2);" ...CASINO' ": FOR M=1
TO 50 STEP 7: BEEP .2-N/100,M:
NEXT M: NEXT N
5330 DIM J$(11,13): PAUSE 30:
GO SUB 8500: PRINT AT 5,5;
PAPER 5;"SLOT MACHINE"
5340 LET J$(1)="{2SG8} {3SG8}
{2SG8}"
5345 LET J$(2)="{13SG8}"
5350 LET J$(4)="{2SG8}
{2SG8}"
5355 LET J$(9)="{5SG8} {5SG8}"

```

```

5360 LET J$(3)=J$(2): LET J$(5)=J$(4)
: LET J$(6)=J$(4): LET J$(7)=J$(
2): LET J$(8)=J$(2): LET J$(10)=
J$(2): LET J$(11)=J$(4)
5370 FOR N=1 TO 11: PRINT AT 6+N,6;
PAPER 7;J$(N): NEXT N: FOR N=10
TO 15: PRINT AT N,20; PAPER 6;"
{SG5}": NEXT N: PRINT AT 15,19;
PAPER 6;"{SG8}"
5380 PRINT AT 5,21; PAPER 4;"PUNTI ";
AT 8,23; PAPER 7;" 2 ";AT 9,23;"
--A"
5390 PRINT AT 11,23;" 4 ";AT 12,23;"-
BB";AT 14,23;" 8 ";AT 15,23;"CCC
"
5400 LET WT=0: LET TU=1: LET GO=3+
INT (RND*3): PRINT AT 17,8;
PAPER 2; INK 7; BRIGHT 1;"~ 0.1
OGNI PROVA";AT 10,8; PAPER 1;"
HAI ";AT 11,8;" ";GO;"
";AT 12,8;" PROVE "
5410 INPUT "PREMI ENTER PER GIOCARE "
; LINE Z$: LET W=0
5420 PRINT AT 7,8; PAPER 3; INK 7;
BRIGHT 1;"No. ";AT 7,14;" ";TU;
" "
5430 LET NN=20+INT (RND*10): FOR N=1
TO NN: LET FMA=(144+INT (RND*3)
): LET FMB=(144+INT (RND*3)):
LET FMC=(144+INT (RND*3))
5440 FOR M=10 TO 12: PRINT AT M,8;
PAPER 5;" " : NEXT M
5450 PRINT AT 11,9; INK 3;CHR$ FMA;
AT 11,12; INK 1;CHR$ FMB;AT 11,1
5; INK 2;CHR$ FMC: NEXT N
5460 IF FMC=144 THEN LET W=2
5470 IF FMB=145 AND FMC=145 THEN
LET W=4
5480 IF FMA=146 AND FMB=146 AND FMC=1
46 THEN LET W=8
5490 PRINT AT 15,12;W: LET TU=TU+1:
LET WT=WT+W: IF TU<>GO+1
THEN BEEP .5,TU: GO TO 5410
5495 BEEP .5,TU: PRINT AT 15,12;W
5500 GO SUB 8650: GO SUB 8500:
PRINT AT 6,4;"HAI GUADAGNATO ";
WT;" PUNTI";AT 8,5;"SPENDENDO ~
";GO/10: LET J(PL)=J(PL)+WT:
LET C(PL)=C(PL)-(GO/10)
5590 GO SUB 8650: GO SUB 1000: LET TI
=0: GO TO 8600
5600 REM ### FERROVIA ###
5610 GO SUB 8500: LET DSL=0: GO SUB 5
620: GO TO 5700
5620 DIM J$(6,17)
5630 LET J$(1)=" {SG8} {SG8}
{2G3}{G7}"
5635 LET J$(2)=" {SG8} {SG2}
{SG1} {2SG8} {G5}"
5640 LET J$(3)=" {12SG8} {G5}"
5645 LET J$(4)=" {G4}{12SG8}{SG5}{G5}"
5650 LET J$(5)=" {12SG8}{SG5}{G5}"
5655 LET J$(6)=" {SG3}{SG8}{3SG3}
{SG8}{3SG3}{5SG8}"

```

```

5660 FOR N=1 TO 6: PRINT AT 8+N,5;
PAPER 6;J$(N): NEXT N
5670 FOR N=5 TO 13: CIRCLE 80,54,N:
CIRCLE 120,54,N: NEXT N: FOR N=
50 TO 53: PLOT 75,N: DRAW 56,0:
NEXT N
5680 PRINT AT 17,4; PAPER 6;"{3G3}
{G7}{SG4}{3G3}{G7}{SG4}{3G3}
{G7}{SG4}{3G3}{G7}{SG4}{3G3}"
5690 FOR N=7 TO 17: PRINT AT N,26;"
{SG8}": NEXT N: PRINT AT 8,22;
INK 2;"{SG8}"; INK 7;"{SG8}";
INK 2;"{2SG8}": RETURN
5700 PRINT AT 5,7;" IL TRENINO ";
AT 6,7;"& LA FERROVIA": GO SUB 8
650: GO SUB 5710: GO TO 5720
5710 FOR N=4 TO 6: PRINT AT N,4;
PAPER 6;"
": NEXT N: RETURN
5720 LET MI=15+INT (RND*10): PRINT
AT 5,4;"LA FERROVIA DISTA";AT 6,
4;MI;" MIGLIA";AT 7,4;"IL TRENO
COSTA ~2": INPUT "VUOI ANDARE ("
; INVERSE 1;"S"; INVERSE 0;"/";
INVERSE 1;"N"; INVERSE 0;") ";
LINE Z$: IF Z$="N" OR Z$="n"
THEN RETURN
5730 GO SUB 1000: IF LO=1 THEN LET T
I=0: RETURN
5735 PRINT AT 7,4; PAPER 6;"
"
5740 LET C(PL)=C(PL)-2: IF RND>.6
THEN GO TO 5780
5750 GO SUB 5710: PRINT AT 4,4;"NON C
I SONO TRENI ";AT 5,4;"A VA
PORE IN PARTENZA ";AT 6,4;"PER
DI 3 PUNTI";AT 7,4; PAPER 6;"
": LET J(PL)=J(PL)
-3
5760 GO SUB 8650: IF RND>.9 THEN
GO SUB 5710: PRINT AT 4,4;"E OG
GI NEPPURE " : LET DSL=1
5765 IF DSL=0 THEN PRINT AT 4,4;
PAPER 6;"
"
5770 PRINT AT 5,4;"I TRENI DIESEL PAR
TONO ";AT 6,4;"PUNTI ";-3*(DSL=1
)+4*(DSL=0);" " : LET J(PL)=
J(PL)-3*(DSL=1)+4*(DSL=0):
GO TO 5790
5780 GO SUB 5710: PRINT AT 4,4;"SEI F
ORTUNATO: I TRENI";AT 5,4;"A VAP
ORE OGGI PARTONO";AT 6,4;"PUNTI
5 " : LET J(PL)=J(PL)+5
5790 LET FL=1: IF RND<.8 THEN
GO TO 5850
5800 GO SUB 8650: GO SUB 5710:
PRINT AT 4,10; PAPER 2; INK 7;
FLASH 1; BRIGHT 1;" INCIDENTE "
5810 DIM S$(3,9): LET S$(3)="ROTTO":
LET S$(2)="DISTORTO": LET S$(1)
="GRAFFIATO": LET Q=1+INT (RND*3
)
5820 PRINT AT 5,4;"SEI CADUTO ED HAI"
;AT 6,4;S$(Q);" LA CAVIGLIA":

```

```

LET LG=3+2*(Q=2)+7*(Q=3)
5830 GO SUB 8650: GO SUB 5710:
PRINT AT 5,4;"HAI PERSO ";LG;"
PUNTI";AT 4,10; PAPER 6;"
": LET J(PL)=J(PL)-LG: LET FL
=0
5850 IF RND>.5 THEN GO SUB 8650:
GO SUB 5710: PRINT AT 5,4;"HAI
VISTO IL FAMOSO ";AT 6,4;"TRE
NO DEI SOGNI:PUNTI 4": LET J(PL)
=J(PL)+4: LET FL=0
5860 IF RND>.92 THEN GO SUB 8650:
GO SUB 5710: PRINT AT 5,4;"TI I
NVITANO SULLA ";AT 6,4;"LOC
OMOTIVA: PUNTI 8 ": LET J(PL)
=J(PL)+8: LET FL=0
5870 IF FL THEN GO TO 5790
5900 GO SUB 8650: GO SUB 1000: LET TI
=0: GO SUB 8600: RETURN
5990 REM ### SAFARI PARK ###
6000 LET MI=25+INT (RND*10): GO SUB 6
010: GO TO 6080
6010 DIM J$(9,16)
6020 LET J$(1)=" {3SG8}"
6025 LET J$(2)=" {5SG8}"
6030 LET J$(3)=" {13SG8}"
6035 LET J$(4)=" {G5}{13SG8}"
6040 LET J$(5)=" {SG8} {12SG8}"
6045 LET J$(6)=" {G5}{SG5} {12SG8}"
6050 LET J$(7)=" {SG8} {SG8} {SG8}
{SG8} {SG8}"
6055 LET J$(8)=" {SG8} {SG8}
{SG8} {SG8}"
6060 LET J$(9)=J$(8)
6070 LET SUB=6070: FOR N=1 TO 9:
PRINT AT N+6,5;J$(N): NEXT N:
PRINT AT 5,8; PAPER 4;"SAFARI P
ARK": RETURN
6080 PRINT AT 7,19;"ENTRATA";AT 8,21;
"~ 2.50";AT 17,4;"PER ARRIVARCI
";MI;" MIGLIA"
6090 INPUT "VUOI ANDARCI (";
INVERSE 1;"S"; INVERSE 0;"/";
INVERSE 1;"N"; INVERSE 0;") ";
LINE Z$: IF Z$="N" OR Z$="n"
THEN RETURN
6100 GO SUB 1000: IF LO=1 THEN LET T
I=0: RETURN
6110 LET C(PL)=C(PL)-2.5
6120 DATA "LEONI","TIGRI","ELEFANTI",
"SCIMMIE","GIRAFFE","RINOCERONTI
","CAMMELLI","BUFALI","LUPI","ZE
BRE","GNU"
6130 LET AN=0: RESTORE 6120: GO SUB 8
650: GO SUB 8500: FOR N=1 TO 11:
READ Q$: IF RND<.3 THEN NEXT N
6140 PRINT AT 4,4;"NELLA RISERVA VEDI
";AT 5,4;"I SEGUENTI ANIMALI";
AT 16,4;"PUNTI 2";AT 17,4;"PER O
GNUNO"
6150 IF N>4 AND FWD<0 AND 1+INT (RND*
50)<=5 THEN GO TO 6300
6160 PAUSE 80: PRINT AT N+5,15;Q$:
LET J(PL)=J(PL)+2: LET AN=AN+2:
PRINT AT 15,22;"PUNTI";AT 16,24

```

LOAD

Gita turistica

```

;AN: NEXT N: PAUSE 100
6170 LET FL=1: GO SUB 8500: GO SUB 60
70
6180 IF RND>.8 THEN GO SUB 8650:
PRINT AT 4,4;"CADI NELLA BUCA D
EL ";AT 5,4;"LEONE MARINO. P
UNTI -3": LET J(PL)=J(PL)-3:
LET FL=0
6190 IF RND>.8 THEN GO SUB 8650:
PRINT AT 4,4;"MANGI UN INCREDIB
ILE ";AT 5,4;"SUPER-GELATO. P
UNTI 2 ": LET J(PL)=J(PL)+2:
LET FL=0
6200 IF RND>.7 THEN GO SUB 8650:
PRINT AT 4,4;"VIENI PUNTO DA UN
A ";AT 5,4;"VESPA. PERDI 4
PUNTI ": LET J(PL)=J(PL)-4:
LET FL=0
6210 IF FL THEN GO TO 6180
6270 GO SUB 8650: GO SUB 1000: LET TI
=0: LET FWD=FWD-1: GO TO 8600
6300 FOR N=8 TO 13: PRINT AT N,4;
PAPER 2; INK 7; BRIGHT 1;
FLASH 1;"
": NEXT N
6310 PRINT AT 10,7; PAPER 1; INK 7;"
D I S A S T R O ": GO SUB 8650
6320 PRINT AT 9,4;"SEI ENTRATO NELLA
";AT 10,4;"GABBIA DELLE BEL
VE !!! ";AT 11,4;"PREMI ENTER PE
R SAPERE ";AT 12,4;"QUANTI PUNTI
PERDI ! "
6330 PAUSE 0: BEEP .05,0: FOR N=1
TO 90: NEXT N: IF INKEY$<>"
THEN GO TO 6330
6340 GO SUB 8500: PRINT AT 15,8;"PERD
I 3+DADO": GO SUB 8430: PRINT
AT 15,8; PAPER 4;"PERDI ";3+D;"
PUNTI": LET J(PL)=J(PL)-D-3:
6350 GO SUB 8650: GO SUB 1000: LET TI
=0: LET FWD=4: GO TO 8600
6490 REM ### MOVIMENTO PEZZI ###
6500 IF INKEY$<>" THEN GO TO 6500

6510 PRINT AT V(PL),H(PL);" ": BEEP .
3,D: LET P(PL)=P(PL)-.25: IF P(P
L)<=.009 THEN GO SUB 1020:
LET LO=0
6520 FOR N=1 TO D: IF V(PL)=1 THEN
LET H(PL)=H(PL)+3: IF H(PL)>28+
(PL=2) THEN LET V(PL)=V(PL)+(H(
PL)-(28+(PL=2))): LET H(PL)=28+(
PL=2): LET D=D-N: GO TO 6540
6530 NEXT N
6540 FOR N=1 TO D
6550 IF V(PL)>1 AND H(PL)=28+(PL=2)
THEN LET V(PL)=V(PL)+3: IF V(P
L)>19 THEN LET H(PL)=H(PL)-(V(P

```

```

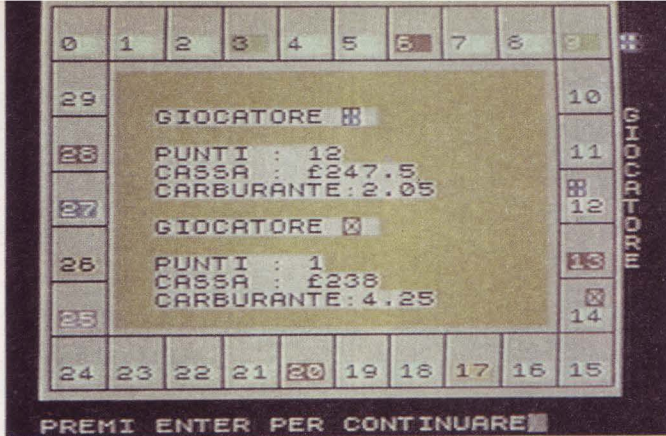
L)-19): LET V(PL)=19: LET D=D-N:
GO TO 6570
6560 NEXT N
6570 FOR N=1 TO D
6580 IF V(PL)=19 AND H(PL)<28+(PL=2)
THEN LET H(PL)=H(PL)-3: IF H(P
L)<1+(PL=2) THEN LET V(PL)=V(PL
)+(-1+H(PL)-(PL=2)): LET H(PL)=1
+(PL=2): LET D=D-N: GO TO 6600
6590 NEXT N
6600 FOR N=1 TO D
6610 IF V(PL)<18 AND H(PL)=1+(PL=2)
THEN LET V(PL)=V(PL)-3: IF V(P
L)<1 THEN LET H(PL)=H(PL)+(1-V(
PL)): LET V(PL)=1: LET D=D-N:
GO TO 6500
6620 NEXT N
6630 PRINT AT V(PL),H(PL); INK PL;
CHR$(144+(PL=2))
6650 LET SUB=ATTR (V(PL)+1,H(PL)):
IF SUB<>104 THEN GO SUB 8500
: GO SUB SUB*50: LET LO=0
6660 LET PL=PL+(PL=1)-(PL=2):
GO SUB 8500: GO SUB 8510:
PRINT AT 2+18*(PL=2),31; INK 1+
(PL=2);CHR$(144+(PL=2)):
RETURN
7990 REM #ALLESTIMENTO SCHERMO#
8000 DATA 4,28,148,172,4,28,244,220,5
,0,5,0,5,0,4,0,5,0,5,0,2,7,5,0,5
,0,4,7,5,0,5,0,5,0,5,0,2,7,5,0,5
,0,6,0,5,0,5,0,5,0,2,7,1,7,6,0,3
,7,5,0,5,0,7,0,2,7,5,0
8110 RESTORE 8000: FOR N=1 TO 4:
READ A: PLOT 4,A: DRAW 240,0:
NEXT N
8120 FOR Y=4 TO 172 STEP 24
8130 PLOT 4,Y: DRAW 24,0: PLOT 220,Y:
DRAW 24,0: NEXT Y
8140 FOR N=1 TO 4: READ A: PLOT A,4:
DRAW 0,168: NEXT N
8150 FOR X=4 TO 248 STEP 24
8160 PLOT X,4: DRAW 0,24: PLOT X,148:
DRAW 0,24: NEXT X
8170 GO SUB 8500
8180 LET NA=0: LET NB=34: LET NC=29:
LET ND=5
8190 FOR L=1 TO 2: FOR N=1 TO 28
STEP 3: IF L=1 THEN READ A:
READ B: PRINT AT 2,N; PAPER A;
BRIGHT 1; INK B;NA;" "
8200 IF L=2 THEN READ A: READ B:
PRINT AT 20,N; PAPER A;
BRIGHT 1; INK B;NB
8210 LET NA=NA+1: LET NB=NB-1: NEXT N
: NEXT L
8220 FOR L=1 TO 2: FOR N=5 TO 17
STEP 3: IF L=1 THEN READ A:
READ B: PRINT AT N,1; BRIGHT 1;
PAPER A; INK B;NC
8230 IF L=2 THEN READ A: READ B:
PRINT AT N,28; PAPER A;
BRIGHT 1; INK B;ND
8240 LET NC=NC-1: LET ND=ND+1: NEXT N
: NEXT L
8250 GO SUB 8510

```

```

8260 PRINT AT 1,1; INK 1;"A";AT 1,2;
INK 2;"B"
8270 PRINT AT 2+17*(PL=2),31; INK 1+(
PL=2);CHR$(144+(PL=2))
8300 REM ##### DADI #####
8310 DIM A$(6,3): DIM B$(6,3): DIM C$(
6,3)
8320 LET A$(1)=" "
8330 LET B$(1)=" C "
8340 LET C$(1)=A$(1)
8350 LET A$(2)="C "
8360 LET B$(2)=A$(1)
8370 LET C$(2)=" C"
8380 LET A$(3)=A$(2): LET B$(3)=B$(1)
: LET C$(3)=C$(2)
8390 LET A$(4)="C C": LET B$(4)=A$(1)
: LET C$(4)=A$(4)
8400 LET A$(5)=A$(4): LET B$(5)=B$(1)
: LET C$(5)=A$(4)
8410 LET A$(6)="CCC": LET B$(6)=A$(1)
: LET C$(6)=A$(6)
8420 LET D$="{SG4}{3G3}{G7}": LET E$=
"{SG1}{3SG3}{SG2}": RETURN
8425 REM #### STAMPA I DADI ####
8430 LET D=1+INT (RND*6): PRINT AT 5,
5;D$;AT 9,5;E$
8440 FOR N=6 TO 8: PRINT AT N,5;"
{SG5}";AT N,9;"{G5}": NEXT N
8450 PRINT AT 6,11;"PREMI ENTER PER"
;AT 7,11;"LANCIARE IL DADO";AT 8
,11;"LASCIA IL TASTO";AT 9,11;"
PER ARRESTARLO": INPUT LINE Z
$
8460 IF D=7 THEN LET D=1
8470 PRINT AT 6,6;A$(D);AT 7,6;B$(D);
AT 8,6;C$(D)
8480 IF INKEY$="" THEN RETURN
8490 LET D=D+1: GO TO 8460
8495 REM ## PULISCE TAVOLA... ##
8500 FOR Y=4 TO 17: PRINT AT Y,4;
PAPER 6;"
": NEXT Y: RETURN
8505 REM #.& L'ULTIMA COLONNA #
8510 FOR N=0 TO 21: PRINT AT N,31;
PAPER 0;" ": NEXT N
8520 DATA "G","I","O","C","A","T","O"
,"R","E"
8530 RESTORE 8520: FOR N=6 TO 14:
READ R$: PRINT AT N,31;
BRIGHT 1; PAPER 0; INK 6;R$:
BEEP .005,N: NEXT N: RETURN
8590 REM ## MOSTRA I PUNTEGGI ##
8600 GO SUB 8610: GO TO 8620
8610 LET P(PL)=(INT (P(PL)*100))/100:
LET C(PL)=(INT (C(PL)*100))/100
: RETURN
8620 GO SUB 8500: PRINT AT 6,6;"GIOCA
TORE "; INK 1;"A ";AT 8,6; INK 0
;"PUNTI : ";J(1);AT 9,6;"CASSA :
~";C(1);AT 10,6;"CARBURANTE:";P
(1);AT 12,6;"GIOCATORE "; INK 2;
"B ";AT 14,6; INK 0;"PUNTI : ";J
(2);AT 15,6;"CASSA : ~";C(2);
AT 16,6;"CARBURANTE:";P(2)
8650 IF INKEY$<>"" THEN GO TO 8650

```

```
8660 INPUT "PREMI ENTER PER CONTINUAR
E"; LINE Z$: BEEP .2,20:
RETURN
```

```
9000 REM ##### ISTRUZIONI #####
9010 PRINT AT 1,8; PAPER 2; INK 7;
BRIGHT 1;"GITA TURISTICA";
BRIGHT 0; INK 0; PAPER 7;AT 13,
6;"Un attimo, prego...":
GO SUB 9100: BORDER 4
```

```
9020 PRINT AT 3,0;"IL TUO OBIETTIVO C
ONSISTE""NEL REALIZZARE QUANTI
PIU'""PUNTI PUOI PRIMA CHE TU O
IL TUO""AVVERSARIO ESAURIATE I
SOLDI."
```

```
9025 PRINT "VI MUOVETE SULLA TAVOLA D
A""GIOCO TIRANDO UN DADO.""OGN
I GIOCATORE MANTIENE IL SUO""TU
RNO FIN QUANDO IL DADO ""NON VI
ENE PASSATO""ALL'AVVERSARIO."
```

```
9030 PRINT AT 13,0;"E' BUONA NORMA CO
MPRARE IL""CARBURANTE QUANDO PO
TETE.""LA BENZINA AVANZATA AL T
ERMINE""DEL GIOCO VIENE COMUNQU
E""TRADOTTA IN PUNTI."
```

```
9040 PRINT AT 19,0;"IL GIOCATORE 1 E
' : "; INK 1;"A"; INK 0""IL GIO
CATORE 2 E' : "; INK 2;"B"
```

```
9050 GO SUB 8650: RETURN
9100 REM #### SET GRAFICO ####
9110 DATA 255,153,153,255,255,153,153
,255,255,195,165,153,153,165,195
,255,0,0,60,60,60,60,0,0
```

```
9120 DATA 0,0,0,0,5,15,7,15,0,12,28,1
25,255,255,255,255,0,0,0,0,193,2
31,239,255,0,0,0,0,192,192,224,2
40,15,15,31,63,127,63,63,15,240,
224,252,252,248,240,224,240,15,1
5,3,1,0,0,0,0,255,255,255,255,12
4,60,24,16,255,255,223,135,6,4,0
,0,240,224,224,128,0,0,0,0
```

```
9130 RESTORE 9110: FOR N=0 TO 12:
FOR M=0 TO 7: BORDER M: BEEP .0
05,N
```

```
9140 READ A: POKE USR CHR$(97+N)+M,A
: NEXT M: NEXT N: RETURN
```

```
9150 STOP
9900 SAVE "GITA" LINE 10
9910 PRINT AT 10,0; PAPER 5;" RIAVVOL
GI IL NASTRO. INSERISCI ""
<EAR> PER VERIFICARE. ""
FAI PARTIRE IL NASTRO E ""
" PREMI ENTER PER VERIFICARE.
"
```

```
9920 VERIFY "GITA"
```



Dalla grande edicola Jackson

Tutto sull'hobby e home computer

hc

HOME COMPUTER

La rivista del computer in casa. "Prove su strada" di software e programmi per tutti i personal computer.

11 numeri all'anno: L. 3.500 a numero
Abbonamento: solo L. 31.500

VIDEO Giochi

La guida indiscussa al fantastico mondo dei videogames. La più eccitante, divertente, istruttiva rassegna del settore.

11 numeri all'anno: L. 3.000 a numero
Abbonamento: solo L. 31.000

elektor

Il mensile di elettronica venduto in mezzo milione di copie e redatto in 7 lingue. Con articoli su: applicazioni, progettazioni, sperimentazioni, invenzioni.

11 numeri all'anno: L. 3.000 a numero
Abbonamento: solo L. 29.000

strumenti MUSICALI

Il mondo delle 7 note in versione...elettronica. Con test strumentali, novità e analisi del mercato, servizi speciali.

10 numeri all'anno: L. 3.000 a numero
Abbonamento: solo L. 24.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:

Gruppo Editoriale Jackson via Rosellini, 12 - 20124 MI

Desidero ricevere GRATIS un numero della Rivista

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

Gli ultimi saranno i primi?

Giudicalo tu.

Anche se, in ordine di tempo, siamo gli ultimi a uscire con libri "dedicati" ai più diffusi home e personal computer, siamo convinti che il lettore attento ed esigente apprezzerà la qualità dei contenuti e la loro presentazione.

Abbiamo selezionato e tradotto i best-seller mondiali che accolgono quanto di più utile è stato scritto per il tuo computer.

Scegli a colpo sicuro!

Puoi ordinare direttamente i titoli presentati compilando ed inviando il coupon pubblicato oppure acquistarli presso i più qualificati computer shop e le migliori librerie.

CEDOLA DI ORDINAZIONE - LIBRI

Compilare e spedire in busta chiusa a:
J. soft - Viale Restelli 5 - 20124 Milano
tel. 02/6880841-6880842-6880843

Ordino i seguenti libri per un importo totale di L.
+ L. 2.000 come contributo fisso per spese di spedizione

Cod. Cod.
Cod. Cod.
Cod. Cod.

- Contanti allegati
- Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul CCP n° 19445204 intestato a J. soft - Milano
- Pagherò in contrassegno al postino al ricevimento dei volumi

Nome

Cognome

Via

CAP Città Prov.

Se richiesta fattura - codice fiscale

Data

J. soft: libri di qualità

Per Commodore 64

Giochi fantastici per il C 64
Cod. ASOC 001 L. 42.000
(inclusa cassetta)

Un libro pensato e realizzato per il divertimento dell'utente del C 64; contiene 19 giochi di vario genere tutti ampiamente commentati. L'attento studio dei programmi potrà essere vantaggioso per l'apprendimento delle tecniche di programmazione.

Il libro del C 64
Volume 1
Cod. ASOC 010 L. 24.000
(inclusa cassetta)
Il libro del C 64
Volume 2
Cod. ASOC 011 L. 24.000
(inclusa cassetta)

I due libri illustrano a fondo le possibilità del Commodore 64. Un compendio di utili consigli, e quant'altro è necessario per conoscere meglio il proprio calcolatore. Esempi pratici completano l'esposizione in modo chiaro ed esauriente.



C 64: suono e grafica - Volume 1
Cod. ASOC 008 L. 24.000
(inclusa cassetta)
C 64: suono e grafica - Volume 2
Cod. ASOC 009 L. 24.000
(inclusa cassetta)

Due volumi che non possono mancare nella biblioteca dell'utente Commodore. Una miniera di idee e suggerimenti per la programmazione del tuo personal computer. Una vasta serie di programmi esemplificativi guidano il lettore al miglior sfruttamento del Commodore 64.

Per ZX Spectrum

15 giochi grafici per ZX Spectrum
Cod. ASOC 002 L. 24.000
(inclusa cassetta)

Un fantastico volume con 15 entusiasmanti giochi grafici per il tuo ZX Spectrum. Tutti i programmi sono "pronti

all'uso" e ampiamente commentati. I programmi sono scritti sfruttando appieno le capacità del calcolatore ed il loro studio può giovare a chiunque intenda affinare le proprie tecniche di programmazione.



Inoltre, i Paper Book: la raccolta dei programmi suddivisi per macchina di Paper Soft, il primo settimanale di software su carta per il tuo computer.

Paper Book - programmi per Apple II -
Cod. ASOC 005 L. 18.000
Paper Book - programmi per VIC 20 -
Cod. ASOC 003 L. 18.000
Paper Book - programmi per Commodore 64 -
Cod. ASOC 004 L. 18.000
Paper Book - programmi per Texas TI 99/4A -
Cod. ASOC 007 L. 18.000
Paper Book - programmi per ZX Spectrum -
Cod. ASOC 006 L. 18.000



In ogni volume, corredato di cassetta, oltre 40 programmi per il tuo computer: giochi grafici, utility, arcade, avventure una miscela esplosiva di software "pronto all'uso"!

J. soft s.r.l.

Viale Restelli, 5 - 20124 Milano - tel. 02/6880841-6880842-6880843

Racer

di **David Price**
trad. e adatt.
di **Marcello Spero**

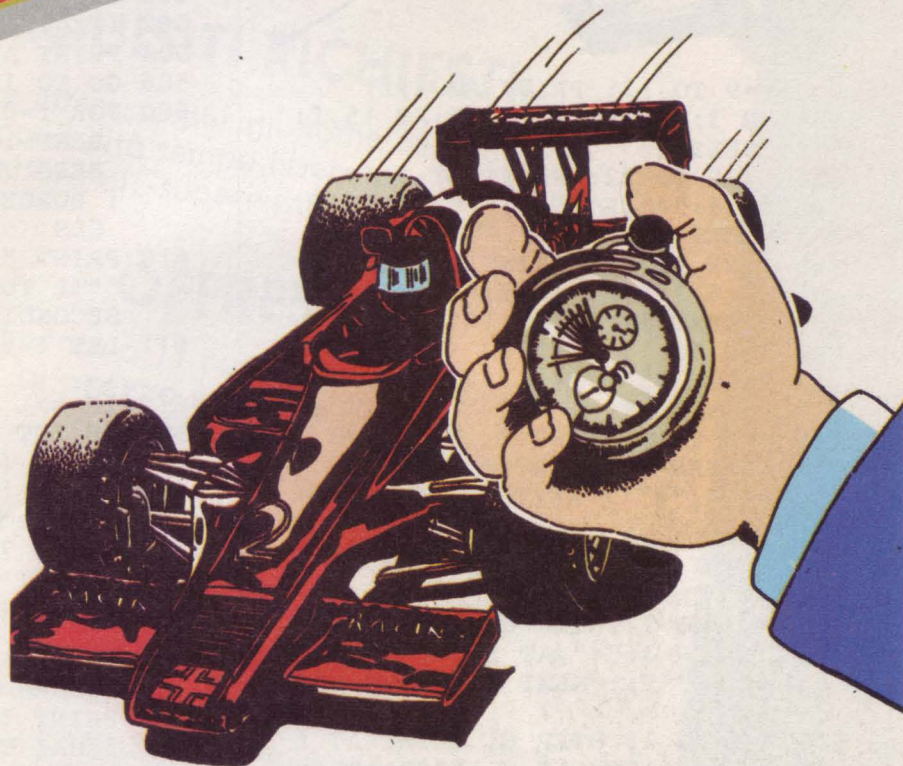
Un arzigogolattissimo percorso da consigliare solo a chi sente di avere la stoffa del pilota.

Fate partire questo programma e vi troverete a misurarvi con le curve e le chicane del circuito, nel tentativo di migliorare il record della pista.

Attenzione alle barriere laterali: il loro urto si rivela sempre letale.

Suono e grafica sono ben utilizzati ed essendo presente la registrazione del record il gioco può diventare avvincente. Istruzioni e controlli sono nel programma.

Pronti? Il record dell'autore è di 32 secondi per l'intero percorso: riuscirete a batterlo? Vedrete che dopo un paio di partite, come è già successo a noi, stenterete a credere che un tempo simile possa essere effettivamente realizzato (per la cronaca, il nostro record è 97, ottenuto con grande fatica...)



Racer

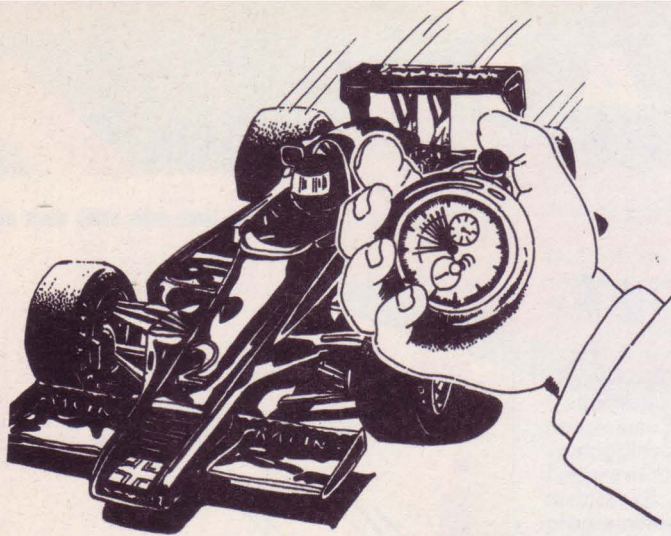
versione per ZX Spectrum 16K/48K

```

1 GO SUB 1000: LET a=2: BORDER 1:
  PAPER 1: CLS
3 PRINT AT 8,6; INK a;"{3SG8}
  {3SG8} {3SG8} {3SG8} {3SG8}
  {SG8} {SG8} {SG8} {SG8} {SG8}
  {SG8} {SG8} {SG8} {SG8}
  {3SG8} {3SG8} {SG8}
  {3SG8} {3SG8}
  {2SG8} {SG8} {SG8} {SG8}
  {SG8} {2SG8}
  {SG8} {SG8} {SG8} {SG8} {3SG8}
  {3SG8} {SG8} {SG8}"
5 IF a=8 THEN BEEP .3,12: BEEP .5
  ,6: BEEP .3,15: GO TO 10
6 LET a=a+1: FOR f=1 TO 4: LET o=.
  0123: BEEP o,2: BEEP o,4: BEEP o
  ,6: BEEP o,8: BEEP o,10: BEEP o,
  12: BEEP o,14: BEEP o,16: BEEP o
  ,2: BEEP o,4: BEEP o,6: NEXT f:
  GO TO 3
10 INK 7: CLS : LET s=0: LET tempo=
  0
  
```

```

15 PRINT AT 3,13;"RACER"" (A)-- S
  INISTRA"," (S)-- DESTRA"" (K)-
  - SU"," (M)-- GIU'"
17 PRINT "" QUESTA E'UNA GAR
  A CONTRO IL TEMPO
  "
20 PRINT "" PREMI UN TASTO PER
  PARTIRE ": PAUSE 0
90 LET q=1: LET la=100: LET giri=1:
  BORDER 1: PAPER 1: INK 9: CLS
100 FOR f=0 TO 31: PRINT AT 1,f;
  PAPER 2;"*";AT 20,f;"*";:
  NEXT f
101 FOR f=1 TO 20: PRINT AT f,0;
  PAPER 2;"*";AT f,31;"*";:
  NEXT f
103 FOR f=3 TO 19: PRINT AT f,3;
  PAPER 2;"*";AT f,13;"*";AT f,19
  ;"*";AT f,28;"*": NEXT f
104 FOR f=2 TO 18: PRINT AT f,6;
  PAPER 6;"*";AT f,16;"*";AT f,25
  ;"*": NEXT f
105 FOR f=7 TO 10: PRINT AT 6,f;
  PAPER 3;"*";AT 12,f;"*";AT 18,f
  ;"*": NEXT f
  
```



```

106 FOR f=9 TO 12: PRINT AT 3,f;
    PAPER 3;"*";AT 9,f;"*";AT 15,f;
    "*": NEXT f
107 FOR f=20 TO 22: PRINT AT 3,f;
    PAPER 4;"*";AT 9,f;"*";AT 15,f;
    "*": NEXT f
108 FOR f=22 TO 24: PRINT AT 6,f;
    PAPER 4;"*";AT 12,f;"*";AT 18,f;
    "*": NEXT f: GO TO 190
120 FOR f=3 TO 15 STEP 6: PRINT AT f
    ,8; PAPER 3;"*";AT f,23; PAPER 4
    ;"*";: NEXT f
121 FOR f=6 TO 18 STEP 6: PRINT AT f
    ,11; PAPER 3;"*";AT f,21;
    PAPER 4;"*";: NEXT f: GO TO 200
140 FOR f=4 TO 16 STEP 6: PRINT AT f
    ,8; PAPER 3;"*";AT f,23; PAPER 4
    ;"*";: NEXT f
141 FOR f=5 TO 17 STEP 6: PRINT AT f
    ,11; PAPER 3;"*";AT f,21;
    PAPER 4;"*";: NEXT f: GO TO 200
160 FOR f=3 TO 18 STEP 6: PRINT AT f
    ,2; PAPER 6;"*";AT f,5;"*";AT f,
    14;"*";AT f,17;"*";AT f,26;"*";
    AT f,29;"*";: NEXT f
161 FOR f=6 TO 18 STEP 6: PRINT AT f
    ,1; PAPER 6;"*";AT f,4;"*";AT f,
    15;"*";AT f,18;"*";AT f,27;"*";
    AT f,30;"*";: NEXT f: GO TO 200
180 FOR f=8 TO 14 STEP 6: PRINT AT f
    ,8; PAPER 3;"*";AT f,23; PAPER 4
    ;"*";: NEXT f
181 FOR f=7 TO 13 STEP 6: PRINT AT f
    ,11; PAPER 3;"*";AT f,21;
    PAPER 4;"*";: NEXT f: GO TO 200
194 POKE 23673,255: POKE 23672,255
196 PRINT AT 20,29; INK 7;"+";AT 20,
    30;"+"
198 PRINT AT 0,0;"TEMPO RECORD ";
    INT tempo;AT 21,0;"TEMPO";AT 21,
    16;"GIRI "
200 PRINT AT 21,21;giri: LET a=19:
    LET b=1
205 LET ti=(256*PEEK 23673+PEEK 2367
    2)/50
208 PRINT AT 21,6;INT ti
210 PRINT AT a,b;"A";AT a,b;"A";AT a
    ,b;"A";AT a,b;" "

```

```

220 LET b=b-(INKEY$="a")+(INKEY$="s"
    ): BEEP .001,3
230 LET a=a-(INKEY$="k")+(INKEY$="m"
    )
240 IF SCREEN$ (a,b)="*" THEN FOR q
    =0 TO 7: PRINT AT a,b; FLASH 1;"
    A": BORDER q: BEEP .09,RND*20:
    NEXT q: GO TO 700
250 IF SCREEN$ (a,b)="+" THEN
    BEEP .5,6: GO TO 500
260 PRINT OVER 1;AT a,b;"A"
300 GO TO 205
500 LET la=la+20: LET giri=giri+1
502 IF giri=6 THEN GO TO 600
504 PRINT AT 21,21;giri
506 GO TO la
600 FOR f=0 TO 7: BEEP .002,RND:
    BEEP .003,RND: BEEP .008,RND:
    BEEP .006,RND: BORDER f: NEXT f
    : BORDER 7: PAPER 7: INK 0:
    CLS
610 PRINT "HAI TERMINATO LA CORSA"
    "'
    "IL TUO TEMPO E' DI ";INT ti;"
    SECONDI"
611 LET s=s+1
612 IF s>1 THEN GO TO 630
615 IF INT (ti)>tempo THEN LET temp
    o=INT (ti)
620 PRINT ""RECORD DELLA PISTA ";
    INT tempo
625 GO TO 730
630 IF INT (ti)<tempo THEN LET temp
    o=INT (ti)
640 GO TO 620
700 BORDER 7: PAPER 7: INK 0: CLS
710 PRINT "HAI URTATO LE BARRIERE"
    "'
    "HAI CORSO PER ";INT ti;" SECON
    DI"
720 PRINT ""HAI COMPLETATO ";giri-
    1;" GIRI""IL RECORD DELLA PIS
    TA E' DI""
    ";tempo;" SEC
    ONDI"
730 PRINT "" UN' ALTRA CORSA (S
    /N)"
731 IF INKEY$="S" OR INKEY$="s"
    THEN GO TO 50
732 IF INKEY$="N" OR INKEY$="n"
    THEN GO TO 2900
733 GO TO 731
1000 FOR a=0 TO 7: READ n: POKE USR "
    A"+a,n: NEXT a: RETURN
1010 DATA 60,60,60,255,255,255,102,10
    2
2900 BORDER 7: PAPER 7: CLS
3000 FOR q=1 TO 6
3003 PRINT AT 10,12; INK q;"CIAO"
3004 LET q=q+1: FOR f=1 TO 4: LET a=.
    012: BEEP a,-4: BEEP a,-6:
    BEEP a,-8: BEEP a,-10: BEEP a,-
    12: BEEP a,-14: BEEP a,-16:
    BEEP a,-2: BEEP a,-4: BEEP a,-6
    : NEXT f: NEXT q
5000 NEW

```



CERCA:

ambosessi di qualsiasi età, residenti in Italia
o all'estero

REQUISITI RICHIESTI:

forte interesse per gli home computer
Sinclair disponibilità tempo libero per appassionante
lettura di "superivista" dedicata

OFFRE:

abbonamento a 11 numeri di SuperSinc
al **prezzo speciale** di
L. 66.000
compresa cassetta
oppure
L. 30.000
per la sola rivista

inviando il coupon pubblicato a fondo pagina otterrete pronta soddisfazione alla vostra richiesta.
Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure
ritagliate il tagliando abbonamenti pubblicato in questa pagina e spedite il tagliando allegando un assegno intestato a:
J. soft - Viale Restelli, 5 - 20124 Milano.

Tagliando abbonamento a SUPERSINC da inviare in busta chiusa a:
J.soft - Viale Restelli 5 - 20124 Milano

- Abbonamento a 11 numeri di SUPERSINC al prezzo speciale di L. 30.000
- Abbonamento a 11 numeri di SUPERSINC + cassetta con tutti i programmi pubblicati al prezzo speciale di L. 66.000

cognome

nome

via

città

cap.

provincia

data

firma

Sinc 3D

di **Martin Clayton**
trad. e adatt.
di **Marcello Spero**

Rotazione, proiezioni ortogonali e assonometrie di qualsiasi oggetto.

Copiate e fate girare il programma, ed otterrete una breve spiegazione del suo funzionamento.

La figura dovrà in ogni caso essere formata da sole linee rette, pur potendo la sua sezione avere qualsiasi numero di lati. Lo stesso VU-3D, forse il miglior programma di disegno a tre dimensioni presente sul mercato, ha problemi con le linee curve.

Le coordinate della figura che volete veder riprodotta andranno poste nell'istruzione DATA di linea 9000, espresse in coordinate relative come per una DRAW. Il punto di partenza sarà sempre il vertice in basso a destra; la prima coppia di coordinate, quindi, sarà espressa in relazione a questo punto.

Nel programma trovate già le coordinate di un esempio; un altro, più semplice, potrebbe essere

9000 DATA -50,0, -10, 20, 40, 20,
20, -40, 1000

Il numero 1000 non ha valore, ma è necessario a segnalare al computer la fine dei dati.

Una volta introdotte le coordinate della vostra figura fate girare il programma. Dopo le istruzioni vi verrà chiesto di specificare la profondità della figura, esprimendola in pixel. Quindi il programma inizia la serie di vedute:

1) rotazione. La figura viene fatta ruotare di 90 gradi, attraverso un numero di vedute specificato da voi.

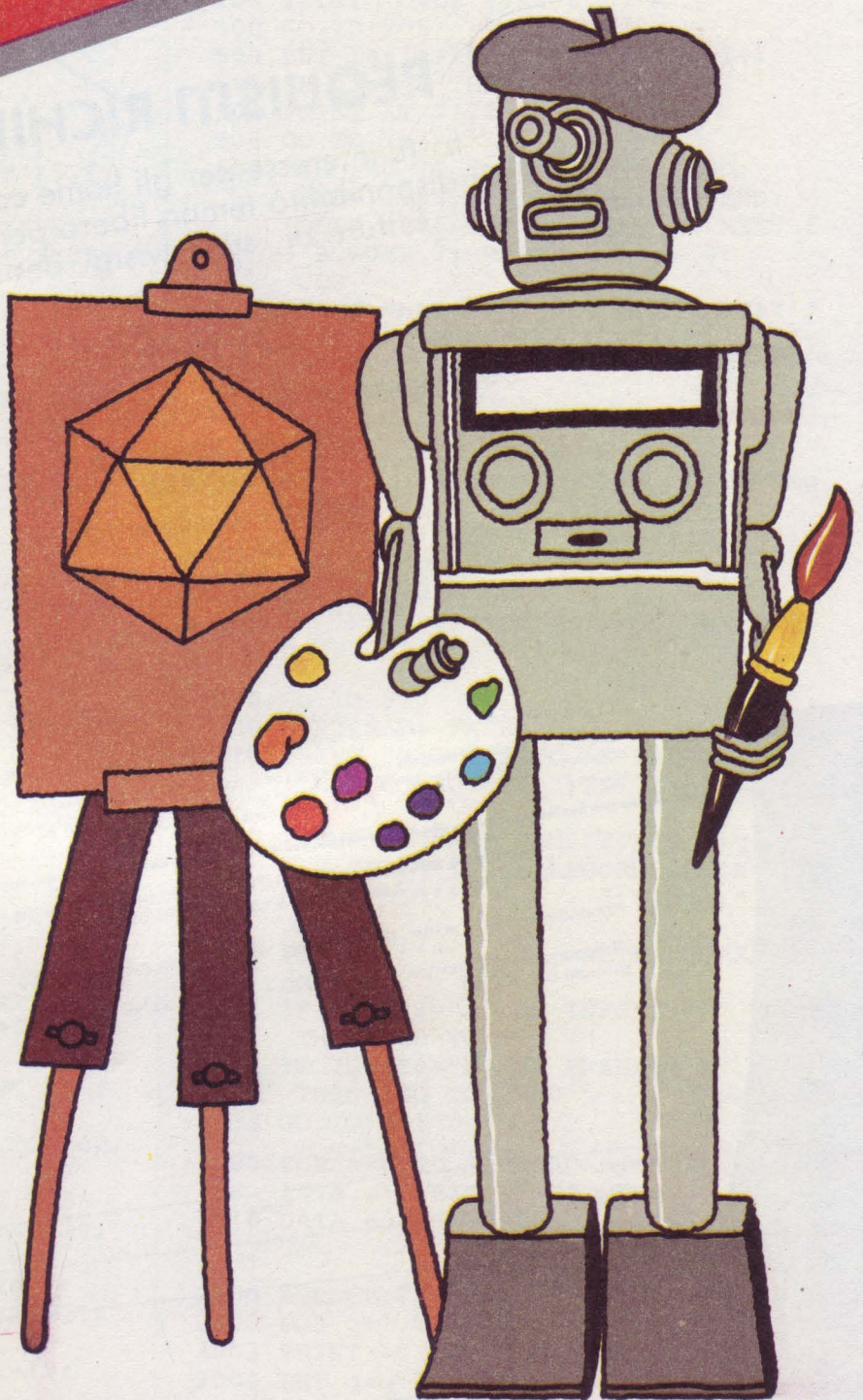
2) proiezione ortogonale. Ben nota agli studenti, si compone di tre vedute: alzata, pianta e laterale, qui disegnate nella loro corretta posizione reciproca.

3) assonometria obliqua. E' un'assonometria 0-45, in cui cioè i piani frontali sono rivolti verso l'osservatore, mentre gli altri piani orizzontali formano con i primi un angolo di 45 gradi.

4) assonometria isometrica. E' un'assonometria 30-30, in cui cioè tutti i piani orizzontali formano un angolo di 30 gra-

di con il piano dell'osservatore. Tutti gli angoli disegnati dal programma sono ottenuti per mezzo di una formula trigonometrica, realizzata dall'autore non senza fatica. Ad ogni stadio del programma viene

offerta la possibilità di ottenere una copia del disegno su stampante. Per comprendere meglio il funzionamento del programma fatelo girare e osservatene il comportamento, grazie all'esempio già incluso.



Sinc 3D

versione per ZX Spectrum 16K/48K

```
10 BORDER 1: PAPER 1: INK 7: CLS :
  RESTORE : GO TO 1000
40 DIM a(100,2): DIM b(100): LET co
nt=0: LET a1=0: LET a2=0: LET a3
=0
50 READ c: IF c=1000 THEN GO TO 61

51 LET a1=a1+c: IF a1<a2 THEN
  LET a2=a1
52 IF a1>a3 THEN LET a3=a1
60 LET cont=cont+1: READ d: LET a(c
ont,1)=c: LET a(cont,2)=d:
  GO TO 50
61 LET largh=a3-a2
69 REM * ROTAZIONE *
70 INPUT "Profondita', in pixel
  (minimo 2, massimo 50)? "
;prof: IF prof<2 OR prof>50
  THEN GO TO 70
72 CLS
75 PRINT AT 0,0;"Rotazione":
  INPUT "Numero di vedute (minimo
  2)? ";ved: LET ved=INT ved:
  IF ved<2 THEN GO TO 75
76 LET ved=ved-1
77 LET pp=(127+largh/2)-a3
80 FOR n=0 TO PI/2+.001 STEP PI/(
  2*ved)
90 FOR m=1 TO cont: LET b(m)=a(m,2)
  *COS n: NEXT m
95 LET a=(INT (((PI/2-n)*(90/PI*2
  ))*100+.5))/100
100 CLS : PRINT "Angolo:";a; " ":
  PLOT pp,70: FOR m=1 TO cont:
  DRAW a(m,1),b(m): NEXT m: IF n=
  0 THEN GO TO 200
150 LET x=prof*SIN n: PLOT pp,70 :
  DRAW 0,-x: FOR m=1 TO cont:
  DRAW a(m,1),b(m): DRAW 0,x:
  DRAW 0,-x: NEXT m
200 INPUT "Copia su ZX Printer? (s/n
  ) "; LINE k$: IF k$="s" THEN
  COPY
220 NEXT n
225 REM * PROIEZIONE ORTOGONALE *
230 LET pp=ABS a2+10
250 CLS : PRINT AT 0,0;"Proiezione o
rtogonale"
260 PLOT pp,70: FOR n=1 TO cont:
  DRAW a(n,1),a(n,2): NEXT n
270 PLOT 170,70: FOR n=1 TO cont:
  DRAW 0,a(n,2): NEXT n: PLOT 170
,70: DRAW prof,0: FOR n=1 TO con
t: DRAW 0,a(n,2): DRAW -prof,0:
  DRAW prof,0: NEXT n
280 PLOT pp,10: FOR n=1 TO cont:
  DRAW a(n,1),0: NEXT n: PLOT pp,
10: DRAW 0,prof: FOR n=1 TO cont
: DRAW a(n,1),0: DRAW 0,-prof:
  DRAW 0,prof: NEXT n
290 INPUT "Copia su ZX Printer? (s/
n) "; LINE k$: IF k$="s" THEN
  COPY
```

```
300 REM * ASSONOMETRIA OBLIQUA *
310 LET x=COS (PI/4)*(prof/2):
  LET pp=120+largh/2
315 CLS : PRINT AT 0,0;"Assonometria
obliqua"
320 PLOT pp,10: FOR n=1 TO cont:
  DRAW a(n,1),a(n,2): NEXT n:
  PLOT pp,10: DRAW x,x: FOR n=1
  TO cont: DRAW a(n,1),a(n,2):
  DRAW -x,-x: DRAW x,x: NEXT n
330 INPUT "Copia su ZX Printer? (s/
n) "; LINE k$: IF k$="s" THEN
  COPY
499 REM * ASSONOMETRIA ISOMETRICA *
500 CLS : RESTORE 9000: PRINT "Asson
ometria isometrica"
520 DIM c(50,3): LET cont=0
525 READ c: IF c=1000 THEN GO TO 60
0
530 LET cont=cont+1: READ d
540 LET c(cont,1)=c*COS (PI/6):
  LET c(cont,2)=d+(-c*SIN (PI/6)
  )
550 GO TO 525
610 PLOT 150,0
620 FOR n=1 TO cont: DRAW c(n,1),c(n
,2): NEXT n
630 LET e=prof*COS (PI/6): LET f=pr
of*SIN (PI/6)
640 PLOT 150,0: DRAW e,f
650 FOR n=1 TO cont: DRAW c(n,1),c(n
,2): DRAW -e,-f: DRAW e,f:
  NEXT n
660 INPUT "Copia su ZX Printer? (s/
n) "; LINE k$: IF k$="s" THEN
  COPY
999 STOP
1000 FOR n=0 TO 7: READ a: POKE USR "
a"+n,a: NEXT n: DATA 48,72,72,48
,0,0,0,0
1005 PRINT TAB 7; INVERSE 1;" DISEGNO
TECNICO "; INVERSE 0
1010 PRINT "'Rotazione di un oggetto
per 90 gradi,
proiezione ortogonale,
assonometria obliqua
ed isometrica."
1020 PRINT "'Mettete i dati relativi
alla vostra figura (formata d
a sole linee rette) nell'istuzi
one DATA di linea 9000, in questo
modo: le coordinate di ciascun
vertice in forma di spostamenti
relativi come per una DRAW, separ
ati da virgole, ed alla fine
il numero 1000."
1021 PRINT "'Iniziate dal vertice in
basso a destra. Per introdurre i
dati, date BREAK ed alterate
la linea 9000."
1050 PRINT #1;"Un tasto per continuar
e": PAUSE 0: GO TO 40
9000 DATA -55,0,0,30,-15,-30,-15,7,-4
,-7,-22,0,20,40,15,-10,30,60,30,
-60,-20,0,-3,5,0,-8,25,0,14,-27,
1000
```

Il tuo settimanale

PAPER soft



Il primo settimanale di software su carta per il tuo personal computer.

Un appuntamento al quale non puoi mancare ogni venerdì in edicola.

Nato in edizione unica, per i più diffusi home e personal computer, dal 1 Aprile,

allo stesso prezzo, saranno in edicola le versioni dedicate

ai prodotti Sinclair (ZX81, ZX Spectrum e QL) e ai

prodotti Commodore (VIC20, C16 e C64).

In più un'edizione esclusiva per gli utenti dei personal

Apple, gli home computer Texas e i nuovissimi sistemi MSX.

Una grossa novità: la possibilità di abbonarsi dal 1 Aprile 1985

al 31 Dicembre 1985 per ricevere 38 numeri del settimanale

comodamente a casa con un notevole risparmio. Chi si abbona

riceverà ogni 15 giorni 2 numeri di PAPERSoft

in anticipo rispetto

all'uscita in edicola.

Non perdere questa occasione!

Compila subito il coupon qui riportato e spediscilo a:

J.soft

**V.le Restelli, 5
20124 MILANO**

EDITRICE 

Tagliando abbonamento a PaperSoft da inviare in busta chiusa a:
J.soft - V.le Restelli, 5 - 20124 MILANO

- Abbonamento a 38 numeri di PaperSoft al prezzo speciale di L. 30.000 anziché L. 38.000
- edizione Sinclair (PS01) edizione Commodore (PC01) edizione Apple/TI99/MSX (PV01)
- contanti allegati assegno allegato n. _____
- ho versato l'importo sul c.c.p. n. 19445204 intestato a J.soft - MILANO

Cognome _____ Nome _____

Via _____ Città _____

CAP _____ Prov. _____ Data _____

Firma _____

Sistema solare

di **Guido Bertolotti**

Con questo programma astronomici dilettanti e... chiunque lo voglia potranno ottenere dai loro ZX81 risultati sorprendenti!

Questo programma si propone di descrivere, prima in forma numerica e poi graficamente, i moti di corpi in orbita intorno al Sole, siano essi pianeti, asteroidi, comete, meteoriti o sonde spaziali.

La completa comprensione del programma richiede alcune conoscenze di geometria analitica: il presente testo è ridotto all'indispensabile, mentre le didascalie delle figure contengono qualche richiamo più preciso sugli algoritmi e sulle formule usate.

Il programma utilizza nei suoi calcoli le tre leggi di Keplero, scoperte nei primi anni del XVII secolo da questo grande astronomo che, insieme a Galileo e Newton, è considerato un pioniere della moderna ricerca scientifica.

La prima legge di Keplero afferma che "le orbite dei pianeti sono ellissi, delle quali il Sole occupa uno dei fuochi". Un'ellisse (v. fig. 1) è simile ad una circonferenza un po' schiacciata. Il "grado di schiacciamento" è detto eccentricità ed è un numero che può variare fra 0 (ellisse non schiacciata) e 1 escluso (massimo schiacciamento).

La seconda legge afferma che "la linea che congiunge un pianeta col Sole (il raggio vettore) copre aree uguali in tempi uguali". Se il pianeta percorre un'orbita non circolare (cioè con eccentricità diversa da 0), la lunghezza del raggio vettore varia: di conseguenza, per coprire la stessa area nello stesso tempo varia anche la velocità del pianeta: essa è massima al perielio (punto di minima distanza dal Sole) e minima all'afelio (punto di massima distanza) (v. fig. 2 e 3).

La terza legge si può enunciare così: "i quadrati dei periodi orbitali dei vari pianeti sono proporzionali ai cubi dei semiassi maggiori delle rispettive orbite" o, in forma più compatta: $T^2 = ka^3$, dove T è il periodo orbitale, k una costante e a il semiasse maggiore dell'orbita.

Vediamo ora il programma: esso com-

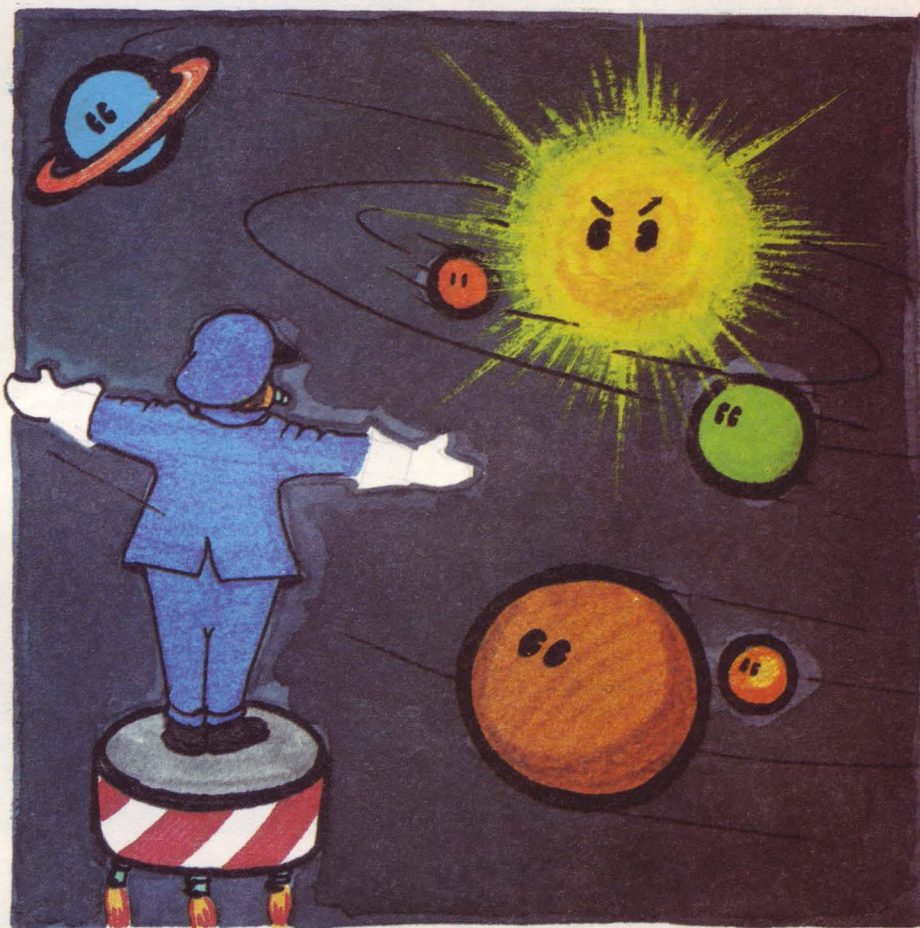


Tabella 1: alcuni dati sui pianeti del sistema solare

Pianeta	Periodo orbitale	Dist. media dal Sole	Eccentricità orbita
Mercurio	88 d	58 Gm	0,186
Venere	224 d	108 Gm	0,007
Terra	1 a	149,7 Gm	0,017
Marte	687 d	228 Gm	0,09
Giove	11 a 308 d	778 Gm	0,048
Saturno	29 a 167 d	1427 Gm	0,056
Urano	84 a	2870 Gm	0,044
Nettuno	165 a	4494 Gm	0,007
Plutone	248 a	5900 Gm	0,248

prende una breve routine in linguaggio macchina, che serve per generare rapidamente un cielo stellato, utilizzato come sfondo nella presentazione e nell'animazione dei pianeti. Per ottenere questa routine è necessario battere il listato 1 e dare RUN. Se appare un cielo stellato con report 0/130, è tutto ok. Cancellate allora tutte le linee, ad eccezione della REM iniziale (che consigliamo vivamente di non EDITare), battete il resto del programma e date RUN 1900 per salvarlo su cassetta.

Il programma parte da solo. Dopo la presentazione esso richiede i dati, che possono essere, a scelta, il periodo orbitale e l'eccentricità dell'orbita, oppure il semiasse maggiore dell'orbita e l'eccentricità (v. tabella 1). I dati numerici devono essere seguiti dai simboli delle loro unità di misura, come del resto appare chiaro dagli esempi di input che fa il computer. È possibile infatti specificare i periodi in anni e/o giorni e le distanze in milioni di chilometri oppure in unità astronomiche (l'unità astronomica, usata per misurare le distanze nel Sistema solare, venne definita come la distanza media fra il centro della Terra e il centro del Sole. Il valore per essa adottato è di 149,504201 milioni di chilometri, anche se, con osservazioni più recenti, si è scoperto che la distanza Terra-Sole è leggermente superiore).

Si passa poi all'output, che fornisce vari dati, calcolati naturalmente secondo le leggi di Keplero, e (dulcis in fundo) all'animazione, che è decisamente la parte più bella del programma. Se provate a introdurre i dati di un corpo che ruota su un'orbita molto eccentrica (per es. l'asteroide Icaro, che ha il semiasse maggiore dell'orbita pari a 161 milioni di chilometri e l'eccentricità di 0,8), lo vedrete chiaramente accelerare quando si avvicina al Sole e rallentare quando se ne allontana, come diretta conseguenza della seconda legge di Keplero. Dobbiamo dire, per la verità, che il moto del pianeta nell'animazione non è calcolato rigorosamente secondo la legge delle aree, ma con un procedimento semplificato (v. fig. 4), che comunque, specie se teniamo conto dei limiti di risoluzione grafica dello ZX81, serve ugualmente ai nostri scopi.

Le opzioni che si possono scegliere durante l'animazione sono: rallentamento del pianeta, fermata, ripartenza, disegno dell'orbita, ritorno alla visione dei dati, COPY dello schermo su carta, introduzione dei dati di un nuovo pianeta.

fig. 1 - I punti di un'ellisse hanno una particolare proprietà: la somma delle distanze di ciascuno di essi dai due fuochi è costante e pari all'asse maggiore:

$$\overline{F_1P} + \overline{PF_2} = \overline{F_1P'} + \overline{P'F_2} = \overline{F_1P''} + \overline{P''F_2} = 2a.$$

L'eccentricità è definita come il rapporto fra la distanza di un fuoco dal centro dell'ellisse e il semiasse maggiore:

Descrizione del programma

Linee

1
60-220
230-790
800-1030
1040-1245
1250-1350
1360-1540
1750-1770
1780-1830
1900-1910

Commento

contiene il linguaggio macchina per generare un cielo stellato
presentazione
input dati
output dati
calcolo e memorizzazione delle posizioni del pianeta per l'animazione
preparazione dello schermo per l'animazione
animazione
subroutine per il calcolo dell'equivalenza unità astronomiche-milioni di chilometri
subroutine per il calcolo della velocità istantanea in perielio e in afelio
SAVE e autostart

Descrizione delle variabili

A semiasse maggiore dell'orbita (in unità astronomiche)
B semiasse minore dell'orbita (in unità astronomiche)
B2 definita alla linea 1110, è il quadrato del semiasse minore di un'orbita di eccentricità E e semiasse maggiore pari a 1 (v. fig. 4)
C distanza di un fuoco dal centro dell'ellisse (in unità astronomiche) (v. fig. 1)
C\$ memorizza il tasto premuto per la scelta di un'opzione variabile di loop, rappresenta l'angolo formato dalla retta con l'asse delle ascisse (v. fig. 4)
D
D\$ utilizzata nella prima parte del programma per l'input dei dati, sotto forma di stringa, contiene in seguito, dalla linea 800 in poi, il nome del pianeta in inverse video.
E eccentricità dell'orbita
F variabile di loop, usata per scopi vari
G nella sezione di input viene utilizzata come puntatore nella stringa D\$ quando essa contiene il periodo orbitale del pianeta; in seguito, durante il calcolo delle posizioni del pianeta, G è il numero d'ordine della posizione e svolge il ruolo di "subscript" dei vettori X() e Y()
L,M,N coefficienti dell'equazione risolvente del sistema per il calcolo dell'intersezione della retta con l'ellisse (v. fig. 4)
M2 definita alla linea 1160, è il quadrato del coefficiente angolare della retta (v. fig. 4)
P è una costante definita alla linea 1070 e contiene il numero di posizioni del pianeta da calcolare per l'animazione. Può essere variata a piacere
P\$ contiene "SI" o "NO" e permette di saltare il calcolo delle posizioni del pianeta, se esso è stato già effettuato
R variabile utilizzata per fornire alle subroutine 1750-1770 e 1780-1830 la distanza del pianeta dal Sole (in unità astronomiche)
S segno, vale -1 o +1 per selezionare l'intersezione sinistra o destra della retta con l'ellisse nel calcolo delle posizioni del pianeta
T periodo orbitale (in anni)
X,Y coordinate del punto d'intersezione della retta con l'ellisse (v. fig. 4)
X(), Y() vettori contenenti le ascisse e le ordinate delle P posizioni successive del pianeta, opportunamente moltiplicate e incrementate per renderle compatibili con la risoluzione grafica dello ZX81

$$e=c/a$$

Da queste due relazioni, e osservando che $\overline{P''F_1} = \overline{P''F_2} = a$, si ricava facilmente l'altra, usata nel programma (linea 710):

$$\text{semiasse minore} = b = \sqrt{a^2 - c^2} = a\sqrt{1 - e^2}.$$

fig. 2 - Seconda legge di Keplero:

se il tempo impiegato dal pianeta per spostarsi da A a B è pari al tempo impiegato per spostarsi da C a D, allora l'area SCD è pari all'area SAB (e viceversa).

fig. 3 - È possibile calcolare, in base alla seconda legge di Keplero, la velocità istantanea in un punto qualunque dell'orbita. Il calcolo è particolarmente semplice per la velocità in perielio e in afelio.

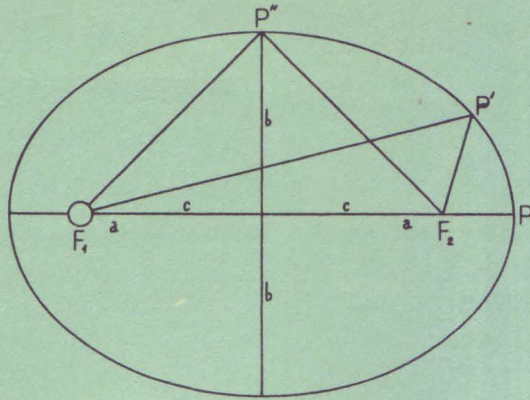


fig. 1

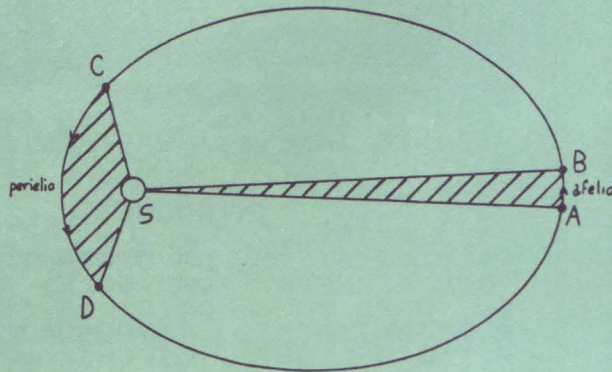


fig. 2

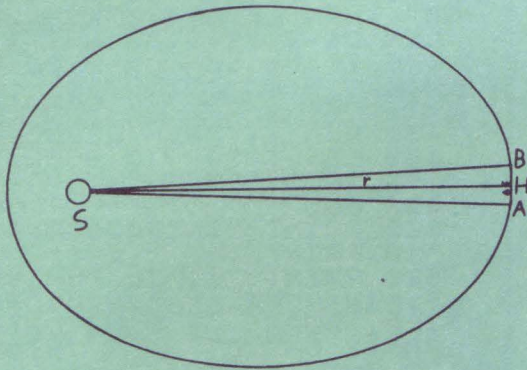


fig. 3

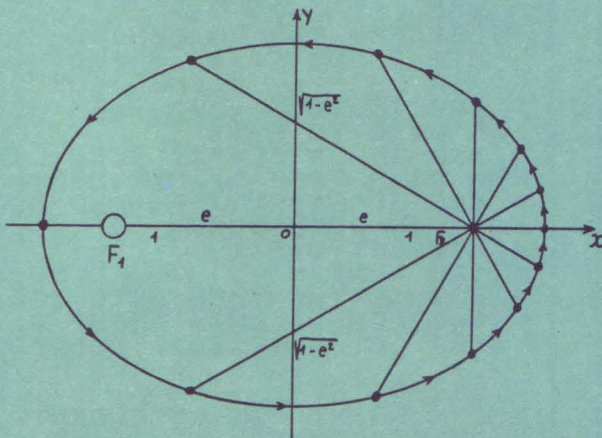


fig. 4

Infatti, se consideriamo un triangolino SAB, di area:

$$A_{(SAB)} = \frac{\pi ab}{n}$$

dove πab è l'area dell'ellisse e n un numero molto grande, la lunghezza della base AB (corta al punto di confondersi con un pezzetto Δs di ellisse) è:

$$\overline{AB} = \Delta s = \frac{2a_{(SAB)}}{r} = \frac{2\pi ab}{rn}$$

dove r è pari all'altezza \overline{SH} .

D'altra parte, il tempo impiegato dal pianeta per percorrere il tratto Δs è, per la seconda legge di Keplero:

$$t = T/n$$

dove T è il periodo orbitale. La velocità istantanea è quindi:

$$v_i = \lim \Delta s / \Delta t = \lim (2\pi abn) / (Trn) = (2\pi ab) / (Tr)$$

$$\Delta r \rightarrow 0$$

$$n \rightarrow +\infty$$

che è la formula usata nella subroutine 1780-1830.

Se esprimiamo a, b, r in unità astronomiche e T in anni, otteniamo la velocità espressa in unità astronomiche/anno. Il programma la trasforma in km/s, moltiplicandola per un'opportuna costante.

fig. 4 - Il programma (linee 1040-1245) calcola le successive posizioni di un pianeta, facendo uso di una legge che veniva utilizzata in luogo della seconda legge di Keplero quando non era necessaria una grande precisione di calcolo.

Si consideri una retta passante per F_2 (il fuoco non occupato dal Sole) e ruotante intorno a F_2 con velocità angolare costante: le successive posizioni del pianeta sono rappresentate da uno dei due punti d'intersezione della retta con l'ellisse.

Il computer esegue i suoi calcoli usando un'ellisse con semiasse maggiore pari ad 1 ed eccentricità E , di equazione:

$$x^2 + y^2 / (1 - E^2) = 1.$$

La retta ha equazione:

$$y = \tan D (x - E)$$

dove D è l'angolo variabile formato dalla retta con l'asse delle ascisse.

Le due equazioni messe a sistema danno la risolvete:

$$(1 - E^2 + \tan^2 D)x^2 - 2(E \tan^2 D)x + (-1 + E^2 + E^2 \tan^2 D) = 0$$

i cui termini fra parentesi sono chiamati nel programma (linea 1170 e seguenti) L, M ed N .

Il programma calcola quindi le coordinate di uno dei due punti d'intersezione, chiamate X e Y , e le memorizza (dopo averle moltiplicate e incrementate per renderle direttamente utilizzabili in istruzioni PLOT e UNPLOT) negli array $X()$ e $Y()$.

Sistema solare

versione per ZX81 16K
caricatore Linguaggio macchina

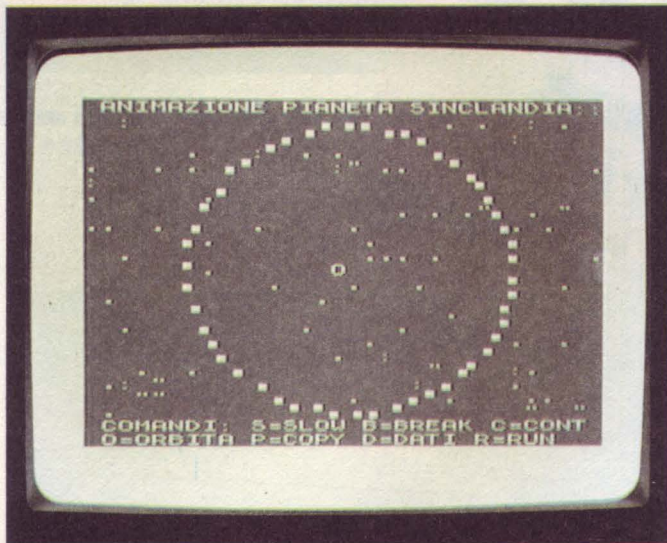
```
1 REM ++++++10+++++20+++
++++30+++++40+++++50
10 REM CODICI IN L M PER CTELO
STELLATO (43 CODICI)
20 FAST
30 LET M$="062,000,050,034,064
,033,000,000,014,024,006,032,030
,120,126,230,082,032,015,030,155
,126,230,012,032,008,030,142,203
,070,040,002,030,139,123,215,038
,016,229,013,032,224,201"
40 LET I=16514
50 LET S=0
60 FOR D=1 TO LEN M$ STEP 4
70 LET V=VAL M$(D TO D+2)
80 POKE I,V
90 LET S=S+V
100 LET I=I+1
110 NEXT D
120 IF S<>3299 THEN PRINT "CONT
ROLLA I CODICI: C^E^ UN ERRO
RE"
130 IF S=3299 THEN RAND USR 165
14
```

Sistema solare

versione per ZX81 16K
programma principale

```
1 REM Y M6RND5 :/42 ACS ?
C^2^?NOT 7( FAST $4 STEP TAN +++
++50
5 REM
10 REM ** SISTEMA SOLARE **
20 REM
30 REM >BY GUIDO BERTELOTTI
40 REM > 12/1984
50 REM
60 REM PRESENTAZIONE
70 SLOW
80 RAND USR 16514
90 PRINT AT 6,2;"":TAB 2
100 PRINT AT 14,14;"":TAB 5;"
110 FOR F=1 TO 200
120 NEXT F
130 CLS
140 PRINT "QUESTO PROGRAMMA CAL
COLA E DISEGNA L'ORBITA DI
UN PIANETA DEL SISTEMA SOLARE
SECONDO LE TRE LEGGI DI KEPLER
O:"
150 PRINT
160 PRINT "1) LE ORBITE DEI
PIANETI SONO ELLISSI DELLE QUA
LI IL SOLE OCCUPA UNO DEI FU
OCHI;"
170 PRINT
180 PRINT "2) LA LINEA CHE C
ONGIUNGE UN PIANETA COL SOLE
COPRE AREE UGUALI IN TEMPI U
QUALI: DI CONSEGUENZA UN PI
ANETA SI MUOVE PIU' VELOC
EMENTE QUANDO E' PIU' VICINO
AL SOLE E PIU' LENTAMENTE QU
ANDO E' PIU' LONTANO;"
190 PRINT
200 PRINT "3) I QUADRATI DEI
PERIODI ORBITALI DEI VARI
```

```
PIANETI SONO PROPORZIONAL
I AI CUBI DEI SEMIASSI MAGG
IORI DELLE RISPETTIVE ORBITE
210 PRINT "CAMBIA C^
CONT"
220 IF INKEY$("<")"C" THEN GOTO 22
0
230 REM TRAP T PAT I
240 LET P$="NO"
250 CLS
270 POKE 16418,2
290 PRINT "QUALI GRANDEZZE CONO
SCI?"
300 PRINT
310 PRINT "1) PERIODO ORBITALE
ED ECCENTRI-CITA" DELL' ORB
ITA"
320 PRINT
330 PRINT "2) SEMIASSE MAGGIORE
DELL' ORBITA (DISTANZA M
EDIA DAL SOLE) ED ECCENTRIC
ITA"
340 PRINT AT 21,0;"":TAB 1;"
350 IF INKEY$="1" THEN GOTO 480
360 IF INKEY$("<")"2" THEN GOTO 35
0
370 CLS
380 PRINT "INTRODUCI:"
390 PRINT
400 PRINT "1) LA DISTANZA MEDIA
DEL PIANETA DAL SOLE. SPECIFI
CA, DOPO IL DATO, SE SI TRATT
A DI UNITA' ASTRONOMICHE (U.
A.) O MILIONI DI CHILOMETRI (G
M) (ES: 228GM);"
410 INPUT D$
420 IF D$(LEN D$-1 TO LEN D$)="
GM" THEN LET D$=STR$(VAL D$( TO
LEN D$-2)/149.504201)+"U.A."
426 IF D$(LEN D$-3 TO LEN D$)<>
"U.A." THEN GOTO 410
440 LET A=VAL D$( TO LEN D$-4)
450 REM >>TERZA LEGGE DI K.<<
460 LET T=SOR (0.99625936*A**3)
470 GOTO 660
490 CLS
500 PRINT "INTRODUCI:"
510 PRINT
520 PRINT "1) IL PERIODO ORBITA
LE, IN ANNI E/O GIORNI. SPECI
FICA, DOPO OGNI DATO, "A"
(ANNI) O "D" (GIORNI) (ES:
1 A 322 D);"
530 INPUT D$
540 LET G=1
550 LET T=0
560 FOR F=1 TO LEN D$
570 IF D$(F)>="." AND D$(F)<="9
" OR D$(F)=" " THEN GOTO 620
580 IF D$(F)="A" THEN LET T=T+U
AL D$(G TO F-1)
590 IF D$(F)="D" THEN LET T=T+U
AL D$(G TO F-1)/365.2564
600 IF D$(F)<>"A" AND D$(F)<>"D
" THEN GOTO 530
610 LET G=F+1
620 NEXT F
630 IF T=0 THEN GOTO 530
640 REM >>TERZA LEGGE DI K.<<
650 LET A=(T**2/0.99625936)**(1
/3)
660 PRINT
670 PRINT "2) L'ECCENTRICITA"
DELL' ORBITA (0<=E<1) (ES: 0
.09)"
680 INPUT E
690 IF E<0 OR E>=1 THEN GOTO 68
0
```



```

700 REM >>PRIMA LEGGE DI K.<<
710 LET B=A*SQR (1-E**2)
720 LET C=A*E
730 PRINT
740 PRINT "3) IL NOME DEL PIANE
TA (MAX 12 CARATTERI) (ES: MA
RTE)."
750 INPUT D$
760 IF LEN D$>12 THEN GOTO 750
770 FOR F=1 TO LEN D$
780 LET D$(F)=CHR$ (CODE D$(F)+
128)
790 NEXT F
800 REM OUTPUT DATI
810 CLS
820 POKE 16418,0
830 PRINT "DATI RELATIVI AL
PIANETA " E ALLA
SUA ORBITA "; AT 1,1; D$
835 PRINT
840 PRINT "SEMIASSE MAGGIORE DE
LL " ORBITA "
842 LET R=A
845 GOSUB 1750
850 PRINT
851 PRINT "SEMIASSE MINORE DELL
" ORBITA "
852 LET R=B
853 GOSUB 1750
855 PRINT
860 PRINT "PERIELIO: DISTANZA D
AL SOLE "
862 LET R=A-C
865 GOSUB 1750
870 GOSUB 1780
880 PRINT "AFELIO: DISTANZA DAL
SOLE "
890 LET R=A+C
900 GOSUB 1750
910 GOSUB 1780
920 PRINT "ECCENTRICITA " DELL
" ORBITA: "; E
930 PRINT
940 PRINT "PERIODO ORBITALE: ";
INT T; " A "; INT (365.2564*(T-INT
T)+.5); " D "
950 PRINT
960 PRINT "SOLE (" ; -C; " U.A.; 0
U.A.) "
970 PRINT AT 23,0; "COM. HERALD
AZIONE P=COPY R=RUN "
980 LET C$=INKEY$
1000 IF C$="A" THEN GOTO 1040

```

```

1010 IF C$="P" THEN COPY
1020 IF C$="R" THEN RUN 230
1030 GOTO 980
1040 REM CALCOLO SEMPLIFICATO DI
E POSIZIONI DEL PIANETA
IN 6 ISTANTI SUCCESSIVI
1050 REM >"SECONDA" LEGGE DI K.<
1055 IF P$="SI" THEN GOTO 1250
1060 FAST
1070 LET P=50
1080 DIM X(P)
1090 DIM Y(P)
1100 LET G=0
1110 LET B2=1-E**2
1120 LET S=-1
1130 FOR D=.51*PI TO 2.5*PI STEP
2*PI/P
1140 LET G=G+1
1150 IF G>P/2 THEN LET S=1
1160 LET M2=ABS TAN D**2
1170 LET L=B2+M2
1180 LET M=E*M2
1190 LET N=M2-B2*M2-B2
1200 LET X=(M+S*SQR (M**2-L*N))/
L
1210 LET Y=TAN D*(X-E)
1220 LET X(G)=INT (X*20+32.5)
1230 LET Y(G)=INT (Y*20+20.5)
1240 NEXT D
1245 LET P$="SI"
1250 REM PREPARAZ. SCHERMO
PER ANIMAZIONE
1260 REM
1270 FAST
1275 CLS
1280 RAND USR 16514
1290 REM >>PRIMA LEGGE DI K.<<
1300 PRINT AT 11, (-20*E+32)/2; "0
"
1310 PRINT AT 0,1; "ANIMAZIONE PI
ANETA "; D$
1320 PRINT AT 22,0; "COMANDI S=
SLOW B=BREAK C=CONT O=ORBITA P=
COPY D=DATI R=RUN "
1330 FOR F=1 TO P
1340 PRINT AT INT ((43-Y(F))/2),
INT (X(F)/2); " "
1350 NEXT F
1360 REM ANIMAZIONE
1370 SLOW
1380 FOR F=1 TO P
1390 UNPLOT X(F),Y(F)
1400 IF INKEY$("<") THEN GOTO 147
0
1410 PLOT X(F),Y(F)
1420 NEXT F
1430 GOTO 1380
1440 FOR D=1 TO P
1450 UNPLOT X(D),Y(D)
1460 NEXT D
1470 LET C$=INKEY$
1480 IF C$="C" THEN GOTO 1410
1490 IF C$="O" THEN GOTO 1440
1500 IF C$="D" THEN GOTO 800
1510 IF C$="P" THEN COPY
1520 IF C$="R" THEN RUN 230
1530 IF C$="S" THEN GOTO 1410
1540 GOTO 1470
1750 REM EQUIVALENZA GM U.A.
1760 PRINT R; " U.A. = "; R*149.50
4201; " GM "
1770 RETURN
1780 REM VELOCITA' ISTANTANEA
1790 REM ALLA DISTANZA S
1800 REM >>SECONDA LEGGE DI K.<<
1810 PRINT "VELOCITA' " ; (2*PI*
A*B)/(T*R)*4.7374192; " KM/S "
1820 PRINT
1830 RETURN
1900 SAVE "SISTEMA SOLARE"
1910 RUN

```

Traduzione numeri

di **Guido Bertolotti**

Una singolare utility per ZX81.

Questo programma insegna al computer come scrivere, in italiano corretto, i numeri interi da zero fino a quasi un miliardo. Per far questo si avvale delle possibilità di elaborazione delle stringhe offerte dal BASIC Sinclair, possibilità queste utilizzate di rado.

La traduzione di numeri dalla forma aritmetica a una lingua parlata, in questo caso l'italiano, non è un problema difficile come potrebbe sembrare. Infatti, le grammatiche delle varie lingue sono (in confronto al resto) molto regolari quando si tratta di formare la parola corrispondente a un numero.

Il programma si avvale di una potente subroutine (linee 8000-9010), che genera la parola corrispondente al numero che le viene fornito.

Questa subroutine, a sua volta, chiama in successione altre subroutine specializzate, ciascuna delle quali pensa a tradurre una parte del numero e ad aggiungerla alla sinistra della parte già tradotta dalle subroutine chiamate prima di lei. La traduzione inizia dall'ultima cifra e continua via via fino alla prima. Se il numero è formato da una sola cifra, il ciclo FOR A.... (linee 8060-8074) chiama una sola subroutine e precisamente la "0-19" (linee 8100-8180). Se il numero è formato da due cifre, il ciclo chiamerà le subroutine "0-19" e "20-90"; se è formato da tre cifre, saranno chiamate la "0-19", la "20-90" e la "100-900", ecc.. Alcune di queste subroutine ne chiamano a loro volta delle altre (la "1000-19000", ad esempio, sbriga una parte del lavoro chiamando la "0-19").

Il programma è suscettibile di vari sviluppi: si potrebbe ad esempio affiancare un'altra subroutine alla 8000-9010, che esegua il lavoro inverso, cioè traduca le stringhe in numeri (magari da una lingua diversa dall'italiano), ottenendo così un programma che traduce numeri fra due lingue; oppure scrivere un programma che genera un numero a caso, lo traduce in due lingue, visualizza una delle traduzioni e richiede l'altra, controllandone poi l'esattezza.



Descrizione del programma

Linee	Commento
10-80	presentazione
110-130	introduzione del numero da tradurre
145-170	visualizzazione del risultato
8000-9010	subroutine per la traduzione del numero in parola
9100-9260	inizializzazione
9300-9310	SAVE e autostart

La subroutine 8000-9010 consiste di un loop (linee 8060-8074) che chiama in successione altre subroutine, e precisamente:

Linee	Nome	Commento
8100-8180	0-19	traducono la cifra delle unità (zerò-nove); se la cifra delle decine è uguale a 1, si salta a 8150 (traducono le ultime due cifre, dieci-diciannove)
8200-8260	20-90	traducono la cifra delle decine (venti-novanta: "dieci" è esclusa, perché è già tradotta da 0-19); se la cifra delle unità è pari a 1 o 8, la variabile D viene posta uguale a 1, per comunicare alla subroutine ADD (li-

8300-8360	100-900
8400-8460	1000-19000
8500-8510	20000-90000
8600-8610	100000-900000
8700-8760	1-19 milioni
8800-8810	20-90 milioni
8900-8910	100-900 milioni
9000-9010	ADD

nee 9000-9010) di troncare l'ultima lettera (questo per far sì che, ad esempio, venga scritto "trentuno" o "novantotto" anziché "trentauno" o "novantaotto").

traducono la cifra delle centinaia. Aggiungono alla sinistra di C\$ "cent" (se la cifra delle decine è un 8) o "cento" (se è diversa da 8). Se la cifra delle centinaia è maggiore di 1, poi, fanno aggiungere (chiamando la subroutine ADD) la corrispondente traduzione in unità (due-nove, per formare duecento-novecento)

traducono la cifra delle migliaia. Aggiungono "mille", se essa è uguale a 1 e le decine e centinaia di migliaia sono pari a 0. Se la cifra delle migliaia è superiore a 1, aggiungono "mila" e chiamano 0-19 (per formare duemila-diciannovemila) chiamano semplicemente 20-90, per aggiungere la traduzione della cifra delle decine di migliaia chiamano 100-900, per aggiungere la traduzione della cifra delle centinaia di migliaia

molto simili alle linee della subroutine 1000-19000: anziché aggiungere "mille" o "mila", aggiungono "un milione" o "milioni" e chiamano 0-19, se la cifra dei milioni è superiore a 1

chiamano 20-90
chiamano 100-900
vengono chiamate dalle subroutine 0-19, 20-90, 100-900 per aggiungere alla sinistra di C\$ la traduzione esatta della cifra corrente.

Aggiungono uno-nove, se P=1, dieci-diciannove, se P=2, ventinove, se P=3. Se D=1, troncano l'ultima lettera, evitando di aggiungerla a C\$

Descrizione delle variabili

- A** variabile di loop, punta, dall'ultimo al primo, i caratteri di N\$. Il loop chiama, una dopo l'altra, le varie subroutine specializzate nella traduzione di una parte del numero
- B,E** variabili di loop, usate nella subroutine di inizializzazione per trasferire le stringhe da M\$ a P\$()
- C** il valore della cifra corrente, cioè VAL N\$(A)
- C\$** la stringa alla quale vengono aggiunti via via i "pezzi" di numero tradotti, finché alla fine contiene il numero scritto in italiano
- D** troncatore: può valere 0 o 1: se vale 1, l'ultimo carattere della stringa da aggiungere alla sinistra di C\$ viene eliminato
- F,G** puntatori della stringa M\$ nella subroutine di inizializzazione
- L** la lunghezza della stringa corrispondente al numero da tradurre (LEN STR\$ N)
- M()** contiene le lunghezze delle stringhe memorizzate in P\$
- M\$** contiene la traduzione delle cifre in italiano. Il suo contenuto, per poter essere utilizzato, viene trasferito in P\$()
- N** il numero da tradurre in italiano
- N\$** la stringa corrispondente a N: STR\$ N. Ad N\$ vengono aggiunti due "0" all'inizio, per impedire un eventuale blocco del programma alle subroutine "1000-19000" o "1-19 milioni"
- P** seleziona i tipi di traduzione di una cifra: ad esempio, se P=1, una cifra C=3 viene tradotta "tre", se P=2, viene tradotta "tredici", se P=3, viene tradotta "trenta"
- P\$** contiene, in ordine, le possibili traduzioni delle cifre. Viene formato durante l'inizializzazione, servendosi delle traduzioni memorizzate in M\$



**LA PRIMA
E UNICA
RIVISTA DI
VIDEOGAMES - COMPUTER
GIOCHI ELETTRONICI**

VideoGiocchi



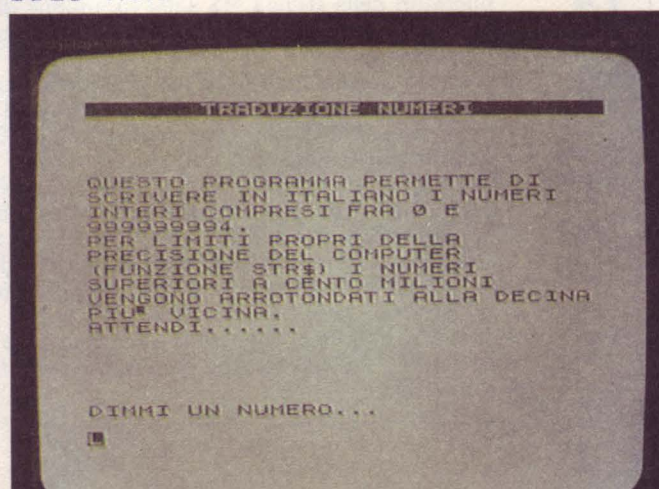
**GRUPPO
EDITORIALE
JACKSON**

Traduzione numeri

versione per ZX81 16K

```
10 REM TRADUZIONE NUMERI
20 REM DI GUIDO BERTOLOTTI
30 REM 1/1985
40 REM
50 PRINT "TRADUZIONE NUMERI"
60 PRINT AT 5,0;"QUESTO PROGRA
MMA PERMETTE DI SCRIVERE IN I
TALIANO I NUMERI INTERI COMPRE
SI FRA 0 E 999999994."
70 PRINT "PER LIMITI PROPRI DE
PRECISIONE DEL COMPU
TER (FUNZIONE STR$) I NU
MERI SUPERIORI A CENTO MI
LIONI VENGONO ARROTONDATI
ALLA DECINA PIU' VICINA."
80 PRINT "ATTENDI....."
90 GOSUB 9100
110 PRINT AT 21,0;"DIMMI UN NUM
ERO..."
120 INPUT N
130 PRINT AT 21,0;"
140 GOSUB 8000
145 CLS
150 PRINT N;" SI SCRIVE IN ITAL
IANO:
160 PRINT C$
170 IF C$="" THEN PRINT "BOH?"
180 GOTO 110
8000 REM TRADUZIONE IN ITALIANO
(STRINGA C$) DEL NUMERO
(VARIABILE N)
8005 LET C$=""
8010 IF N<0 OR N>999999994 OR N<
>INT N THEN RETURN
8020 LET N$=STR$ N
8030 LET L=LEN N$
8040 LET N$="00"+N$
8050 FOR A=L+2 TO 3 STEP -1
8070 LET C=CODE N$(A)-28
8072 GOSUB 100*(L-A)+8300
8074 NEXT A
8080 RETURN
8100 REM 0-19
8110 IF C=0 AND L=1 THEN LET C$="
ZERO"
8120 IF N$(A-1)="1" THEN GOTO 81
50
8125 IF C=0 THEN RETURN
8130 LET P=1
8135 GOSUB 9000
8140 RETURN
8150 REM 10-19
8160 LET P=2
8170 GOSUB 9000
8180 RETURN
8200 REM 20-90
8210 IF C<2 THEN RETURN
8220 LET P=3
8230 IF N$(A+1)="1" OR N$(A+1)="
8" THEN LET D=1
8240 GOSUB 9000
8250 LET D=0
8260 RETURN
8300 REM 100-900
8310 IF C=0 THEN RETURN
8315 IF N$(A+1)<>"8" THEN LET C$
="0"+C$
8320 LET C$="CENT"+C$
8330 IF C=1 THEN RETURN
8340 LET P=1
8350 GOSUB 9000
8360 RETURN
8400 REM 1000-19000
8410 IF N$(A-2 TO A)="000" THEN
```

```
RETURN
8420 IF N$(A-2 TO A)="001" THEN
LET C$="MILLE"+C$
8430 IF N$(A-2 TO A)="001" THEN
RETURN
8440 LET C$="MILA"+C$
8450 GOSUB 8100
8460 RETURN
8500 REM 20000-90000
8505 GOSUB 8200
8510 RETURN
8600 REM 100000-900000
8605 GOSUB 8300
8610 RETURN
8700 REM 1 MILIONE-19 MILIONI
8710 IF N$(A-2 TO A)="000" THEN
RETURN
8720 IF N$(A-2 TO A)="001" THEN
LET C$="UN MILIONE "+C$
8730 IF N$(A-2 TO A)="001" THEN
RETURN
8740 LET C$=" MILIONI "+C$
8750 GOSUB 8100
8760 RETURN
8800 REM 20-90 MILIONI
8805 GOSUB 8200
8810 RETURN
8900 REM 100-900 MILIONI
8905 GOSUB 8300
8910 RETURN
9000 REM AGGIUNGE ALLA SINISTRA
DI C$ L'OPPORTUNO NOME
DELLA CIFRA
9005 LET C$=P$(P,C+1) ( TO M(P,C+
1)-0)+C$
9010 RETURN
9100 REM INIZIALIZZAZIONE
9110 LET M$="ZERO,UNO,DUE,TRE,QU
ATTRO,CINQUE,SEI,SETTE,OTTO,NOVE
, DIECI,UNDICI,DODICI,TREDICI,QUA
TTORDICI,QUINDICI,SEDICI,DICIAS
ETTE,DICIOTTO,DICIANNOVE,ZERO,DI
ECI,VENTI,TRENTA,QUARANTA,CINQUA
NTA,SESSANTA,SETTANTA,OTTANTA,NO
VANTA,CENTO"
9120 DIM P$(3,10,11)
9130 DIM M(3,10)
9140 LET G=1
9150 LET F=3
9160 FOR E=1 TO 3
9170 FOR B=1 TO 10
9180 LET F=F+1
9190 IF M$(F)<>"," THEN GOTO 918
0
9200 LET P$(E,B)=M$(G TO F-1)
9210 LET M(E,B)=F-G
9220 LET G=F+1
9230 NEXT B
9240 NEXT E
9250 LET D=0
9260 RETURN
9300 SAVE "TRADUZIONE NUMERI"
9310 RUN
```

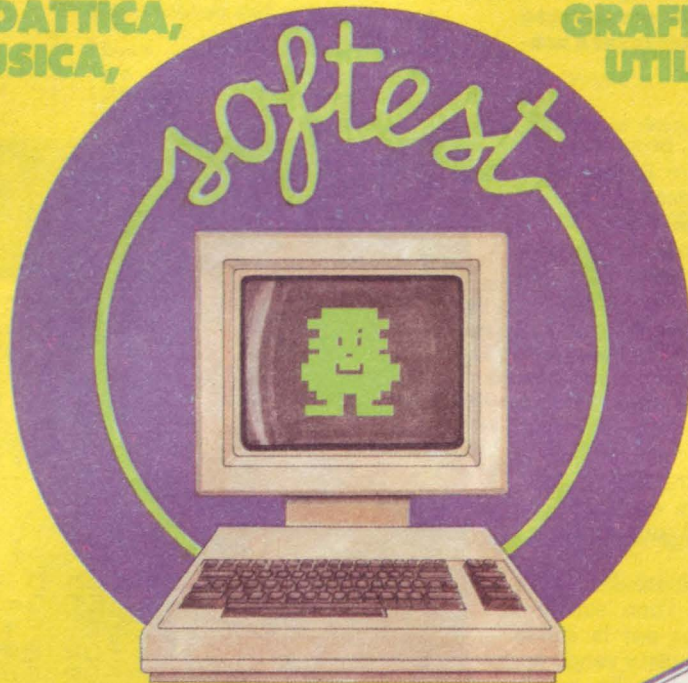


E' IN EDICOLA

IL MEGLIO DI ^{HOME COMPUTER} **hc**

SUPPLEMENTO AL N° 11 DI HC L. 8.000

**IL MEGLIO DEL SOFTWARE
PER IL TUO COMPUTER IN CASA: GIOCHI,
ADVENTURE GAME, GESTIONE DELLA CASA
DIDATTICA, MUSICA, GRAFICA,
UTILITY**



PROVE E RECENSIONI



**IN APPENDICE GUIDA
ALL'ACQUISTO DEL TUO
PERSONAL COMPUTER
SECONDO
LE TUE ESIGENZE**

**UN'OCCASIONE
DA NON PERDERE**



**GRUPPO
EDITORIALE
JACKSON**

San Francisco • Londra • Milano

Music

di **Giovanni Benintende**

Lo Spectrum diventa per voi, questa volta, un utile strumento per comporre e memorizzare musica.

Lo scopo di questo programma è quello di far suonare il computer, impostando le note musicali di un motivo sul pentagramma che appare sul video. Il campo di impiego è quindi molto vasto, basti pensare a tutti coloro che sono capaci di suonare uno strumento musicale senza conoscere la musica e si disperano davanti alla partitura di "Giochi Proibiti" senza poterla suonare, o ai neo-compositori che non sanno trascrivere sul pentagramma le loro idee. La grafica è il punto forte di questo programma: si hanno infatti a disposizione tutte quelle figure essenziali che servono per comporre una partitura musicale.

Tutte le figure sono formate da quattro caratteri grafici, posti l'uno sull'altro (d\$,) che variano anche per la stessa figura, a seconda che questa venga posta sul rigo (d\$=r\$) o nello spazio (d\$=e\$); se infatti non si definisse la stessa figura in due serie di caratteri grafici (l'uno sfalsato di quattro pixel rispetto all'altro), si otterrebbe non più di un'ottava di range, visto che le righe a disposizione sono otto. Usando invece e\$ centrata nello spazio e r\$ centrata nel rigo (4 punti più in basso), la figura può spostarsi in 24 posizioni differenti, portando il range a oltre tre ottave.

Inoltre i 21 caratteri grafici definibili che abbiamo a disposizione non sono più sufficienti a definire tutte le figure, per cui bisogna ridefinire di volta in volta i primi otto: "ABCD" (e\$) e "EFGH" (r\$). Anche il carattere grafico "J" è definito di volta in volta quando si richiamano le alterazioni (diesis, bemolle, punti), mentre i parametri delle stesse (sa, sb) vengono letti contemporaneamente ai dati grafici.

Si può disporre anche delle terzine e delle legature di valore, anch'esse puntualmente disegnate nei particolari, mentre il numero massimo di note ottenibili è di 200, visto che la RAM del 16 K è totalmente utilizzata.



Esempio di uso

(C=computer; O=operatore)

- C:** disegna il pentagramma con la chiave di violino e chiede il tempo
- O:** immettere il tempo stabilito in semiminime/minuto

- C:** disegna il menu e il puntatore (freccia sotto la semiminima)
- O:** si possono usare i seguenti tasti:
 - 5 sposta il puntatore a sinistra
 - 8 sposta il puntatore a destra

- 0 seleziona la figura puntata dalla freccia
 q cancella la nota precedente
 p suona tutte le note definite
C: (dopo che avremo pigiato "0") visualizza la nota prescelta sul pentagramma
O: si possono usare i seguenti tasti:
 6 sposta la nota in basso
 7 sposta la nota in alto
 0 fissa la figura sul pentagramma
C: (dopo aver pigiato "0") visualizza il puntatore sotto l'ultima figura prescelta nel menu e appare in basso a sinistra il numero progressivo della nota da definire
O: ripetere l'operazione di scelta della figura successiva, ecc.

Per ottenere le terzine si fissano le prime due note della terzina e, prima di fissare la terza, si pigia il tasto "3", poi "0" per fissare la nota.

Per la legatura di valore si fissa la prima nota e, prima di fissare la seconda, si pigia il tasto "1", poi "0" per fissare la nota.

Per i bemolle, diesis, punti di valore, prima di fissare la nota si pigia il tasto "1", scompare la nota e appare accanto un bemolle. Si pigia nuovamente l'"1" per andare avanti con il diesis, poi il punto, il punto doppio, il triplo e, daccapo, il bemolle, fino a che uno di questi viene fissato con "0". Dopo il fissaggio riappare la nota e si possono eseguire operazioni dello stesso genere, per esempio per l'uso del punto e del diesis contemporaneamente. Alla fine di tali operazioni la nota deve essere fissata, pigiando "0" per passare a quella successiva.

Descrizione delle variabili

t	tempo di base
j	contatore delle note da definire
f(), g()	durata e tono dei suoni
r\$(), e\$()	composizione della figura sul rigo e nello spazio
d\$()	composizione della figura sul pentagramma
b, h	ascissa e ordinata della figura da definire
i, l	ascissa delle figure e del puntatore nel menu
m	ascissa delle figure di alterazione o ordinata dei tagli addizionali
nt	stabilisce il posizionamento della figura sul rigo (nt=0) o nello spazio (nt=1)
x, s, n, v	contatori
k	dati in lettura e altri usi
sa, sb	parametri in lettura usati nelle alterazioni
a()	dati in lettura relativi alla composizione dei caratteri grafici delle note
t\$, g\$	associate a INKEY\$

Descrizione del programma

Linee	Commento
110	immissione tempo di metronomo
120-140	preparazione dati
150-220	ciclo principale per la definizione delle note
500-540	suona
1000-1070	pentagramma e chiave di violino
2000-2110	preparazione menu
3020-3050	visualizzazione e spostamento del puntatore nel menu
3060	cancella errore
3080	va a suonare
3110-3120	determina la durata della nota
3500-3550	costruisce la figura scelta nel menu
4000	controlla se la figura è una pausa
4040-4070	sposta la figura sul pentagramma
4080	disegna la terzina e ne determina i valori
4085	disegna la legatura e ne determina i valori
4090	dispone il contatore j alla nota successiva
4500	disegna la figura sul pentagramma
4510	disegna i tagli addizionali
5000-5030	determina il tono della nota
5500-5590	alterazioni (bemolle, diesis, punti)
6000-6130	dati relativi alla composizione delle figure
6500-6680	dati per la grafica delle figure
6700	dati per disegnare la chiave di violino
6720	valori delle figure
6730	toni delle figure

Music

versione per ZX Spectrum 16K/48K

```

100 BORDER 1: CLS : GO SUB 1000
110 BEEP .2,25: INPUT "TEMPO (I/min.
) ";t: LET t=60/t
120 DIM f(200): DIM g(200): DIM a(4)
: DIM d$(4): DIM e$(4): DIM r$(4)
)
130 LET l=6: LET i=0: LET j=1: LET h
=6: LET hl=0: LET h2=0: LET b=6:
LET nt=1: LET e$="ABCD": LET r$
="EFGH": LET d$=e$
140 FOR n=0 TO 63: POKE USR "A"+n,0:
NEXT n: GO SUB 2000
150 GO SUB 3000
200 LET j=j+1: LET b=b+3: PRINT AT 2
1,0: BRIGHT 1;j
210 IF b>30 THEN PAUSE 200: LET b=6
: GO SUB 1000
220 GO TO 150
500 PRINT AT 0,0;" ": FOR x=1 TO j

```

```

510 IF g(x)=100 THEN PAUSE 50*f(x):
NEXT x
520 BEEP f(x),g(x)
530 PRINT AT 0,0;x: NEXT x
540 RETURN
1000 FOR x=0 TO 16: PRINT AT x,0;"
":
NEXT x
1010 FOR x=96 TO 131 STEP 8: PLOT 0,x
: DRAW 255,0: NEXT x
1020 PLOT 11,90
1030 RESTORE 6700
1040 FOR x=1 TO 8: READ a,o,c: DRAW a
,o,c: NEXT x
1050 RESTORE 6710
1060 FOR n=0 TO 7: READ k: POKE USR "
I"+n,k: NEXT n
1070 RETURN
2000 FOR x=0 TO 13
2010 RESTORE 6000+x*10
2020 FOR n=1 TO 4: READ a(n): NEXT n
2030 GO SUB 2040: GO TO 2100

```

LOAD

Music

```

2040 FOR s=1 TO 4
2050 RESTORE 6500+10*a(s)
2060 FOR n=0 TO 7
2070 READ k
2080 POKE USR "A"+n+8*(s-1),k: POKE
    USR "E"+4+n+8*(s-1),k
2090 NEXT n: NEXT s: RETURN
2100 LET i=i+2: FOR v=1 TO 4: PRINT
    PAPER 5;AT v+16,i;r$(v);" ":
    NEXT v
2110 NEXT x: RETURN
3010 GO SUB 3020: GO TO 3030
3020 BEEP .005,0: PRINT AT 20,1;
    OVER 1; PAPER 5;"E": RETURN
3030 IF INKEY$="" THEN GO TO 3030
3040 LET t$=INKEY$: GO SUB 3020
3050 LET l=1+(2 AND t$="8" AND l<28)-
    (2 AND t$="5" AND l>2)
3060 IF t$="q" THEN LET j=j-1:
    PRINT AT 21,0;" "; FLASH 1;
    BRIGHT 1;AT 21,0;j: LET b=b-(3
    AND b>6): LET f(j)=0: LET g(j)=
    0: FOR x=0 TO 16: PRINT AT x,b-1
    ;" ";AT x,b-1;("___" AND x>4
    AND x<10): NEXT x: GO TO 3000
3070 IF t$="0" THEN GO TO 3100
3080 IF t$="p" THEN PAUSE 100:
    GO SUB 500
3090 GO TO 3010
3100 RESTORE 6720
3110 FOR n=2 TO 1 STEP 2: READ k:
    NEXT n
3120 LET f(j)=t*k
3500 RESTORE 6000
3510 FOR x=0 TO (1-2)/2
3520 FOR n=1 TO 4: READ a(n): NEXT n
3530 NEXT x
3540 GO SUB 2040
3550 FOR x=7 TO 79 STEP 8: POKE USR "
    A"+x,0: NEXT x
4000 IF l>14 THEN LET d$=e$: LET h=5
    : LET nt=1: LET g(j)=100:
    GO SUB 4500: RETURN
4010 GO SUB 4500
4020 IF INKEY$="" THEN GO TO 4020
4030 LET t$=INKEY$: GO SUB 4500
4040 IF t$="6" AND nt=0 THEN LET h=h
    +(1 AND h<12): LET nt=1
4050 IF t$="7" AND nt=1 THEN LET h=h
    -(1 AND h>1): LET nt=0
4060 IF t$="6" THEN LET d$=r$: LET n
    t=0
4070 IF t$="7" THEN LET d$=e$: LET n
    t=1
4080 IF t$="3" THEN LET f(j)=f(j)*2/
    3: LET f(j-1)=f(j): LET f(j-2)=f
    (j): PLOT 4+8*b,8*(21-h): DRAW -
    48,8*(h-h2),.5: PRINT AT h1-1,b-
    3; OVER 1;3
4085 IF t$="1" THEN LET f(j)=f(j)+f(
    j-1): LET f(j-1)=0: PLOT 4+8*b,8
    *(21-h): DRAW -24,0,1
4090 IF t$="0" THEN GO SUB 4500:
    GO SUB 5000: LET h2=h1: LET h1=
    h: RETURN
4100 IF t$="1" THEN GO SUB 5500
4110 GO SUB 4500: GO TO 4020
4500 BEEP .005,0: FOR x=1 TO 4:
    PRINT OVER 1;AT h+x,b;d$(x):
    NEXT x
4510 LET m=(88 AND h>=9)+(120 AND h<9
    ): LET k=(8 AND h>=9)-(8 AND h<9
    ): FOR v=8*(20-h) TO m STEP k:
    PLOT b*8,v: DRAW 7,0: NEXT v:
    RETURN
5000 RESTORE 6730
5010 FOR x=1 TO 26
5020 READ nota
5030 IF x=29-2*(h+1)+nt THEN LET g(j
    )=g(j)+nota: RETURN
5040 NEXT x
5500 FOR x=10 TO 14: RESTORE 6500+10*
    x: READ sa, sb
5510 FOR n=0 TO 7
5520 READ k: POKE USR "J"+n,k
5530 NEXT n
5540 LET m=b-(1 AND x<12)+(1 AND x>11
    )
5550 PRINT OVER 1;AT h+1,m;"J"
5560 LET g$=INKEY$
5570 IF g$="1" THEN BEEP .005,0:
    PRINT OVER 1;AT h+1,m;"J":
    NEXT x: GO TO 5500
5580 IF g$="0" THEN LET f(j)=f(j)*sa
    : LET g(j)=g(j)+sb: RETURN
5590 GO TO 5560
6000 DATA 0,18,18,18
6010 DATA 1,17,2,18
6020 DATA 3,17,2,18
6030 DATA 3,17,4,18
6040 DATA 3,6,5,18
6050 DATA 3,7,5,18
6060 DATA 3,8,5,18
6070 DATA 9,17,2,18
6080 DATA 9,17,4,18
6090 DATA 9,6,5,18
6100 DATA 9,7,5,18
6110 DATA 9,8,5,18
6120 DATA 16,18,18,18
6130 DATA 18,16,18,18
6500 DATA 0,60,66,66,66,60,0,0
6510 DATA 0,60,66,66,66,124,64,64
6520 DATA 64,64,64,64,0,0,0,0
6530 DATA 0,60,126,126,126,124,64,64
6540 DATA 64,67,76,112,0,0,0,0

```

LOAD

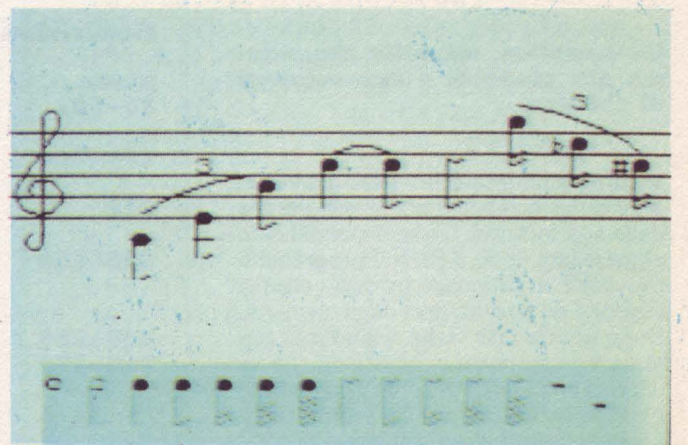
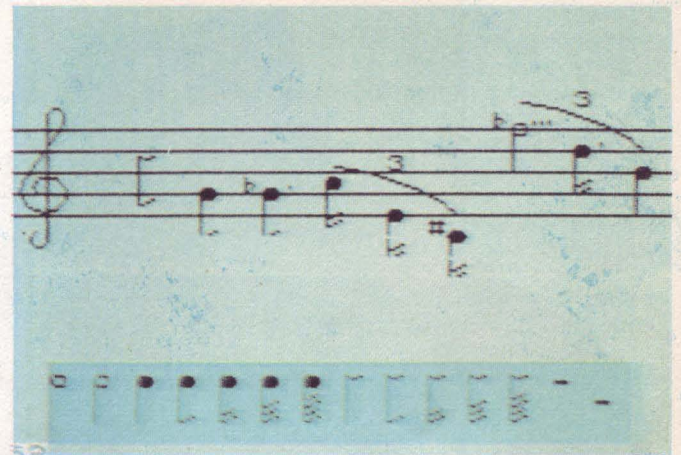
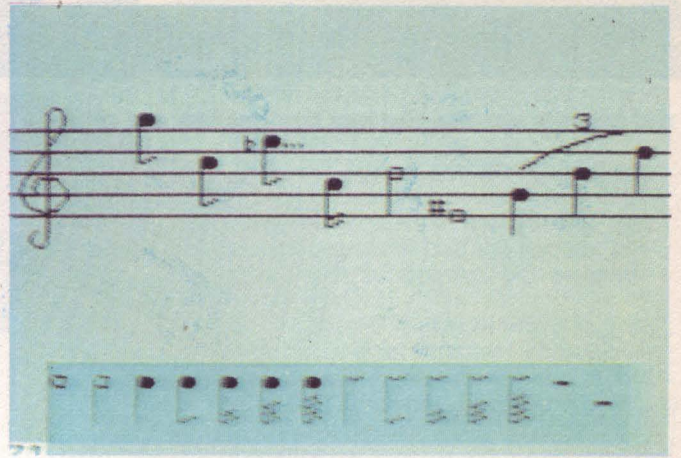
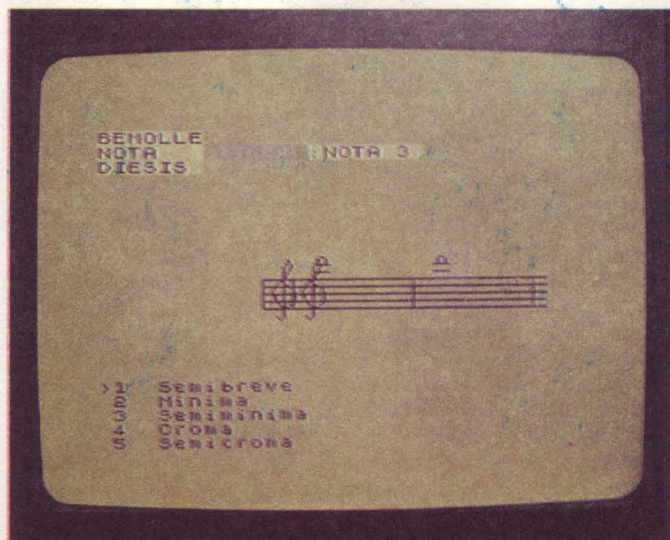
Music

```
6550 DATA 112,67,76,112,0,0,0,0
6560 DATA 64,64,64,64,64,64,67,76
6570 DATA 64,64,64,67,76,112,67,76
6580 DATA 67,76,112,67,76,112,67,76
6590 DATA 0,99,92,64,64,64,64,64
6600 DATA 1,-1,0,32,32,56,36,40,48,0
6610 DATA 1,1,0,36,126,36,36,126,36,0
```

```
6620 DATA 1.5,0,0,0,0,0,16,16,0,0
6630 DATA 1.75,0,0,0,0,0,40,40,0,0
6640 DATA 1.875,0,0,0,0,0,73,73,0,0
6650 DATA 32,36,44,52,36,44,52,4
6660 DATA 0,0,0,62,62,0,0,0
6670 DATA 64,64,64,64,64,64,64,64
6680 DATA 0,0,0,0,0,0,0,0
6700 DATA 4,-4,4,0,47,0,6,0,-3,0,-3,0
      ,-14,-20,0,9,-12,2.7,0,12,3.2,-5
      ,-6,2.5
```

```
6710 DATA 28,62,62,60,32,32,32,32
6720 DATA 4,2,1,1/2,1/4,1/8,1/16,1,1/
      2,1/4,1/8,1/16,2,4
6730 DATA -13,-12,-10,-8,-7,-5,-3,-1,
      0,2,4,5,7,9,11,12,14,16,17,19,21
      ,23,24,26,28,29
```

```
7000 CLS : PRINT AT 10,6; FLASH 1;
      BRIGHT 1;"FERMA IL REGISTRATORE
      ": BEEP 1,20: POKE 23609,50
7010 PAUSE 200: PRINT AT 15,2;"PREMI
      UN TASTO PER INIZIARE": PAUSE 0:
      RUN
7020 CLEAR : SAVE "MUSIC" LINE 7000
```



Pesi molecolari

di Antonio Motta



Un semplice, ma utile programma per studenti e appassionati di chimica.

Il programma calcola il peso molecolare di un composto chimico a partire dalla sua formula. L'uso è semplificato al massimo; dopo il RUN il programma provvede a sistemare in due vettori i simboli ed i pesi atomici degli elementi. Dopo questa fase viene richiesta la formula del composto in questione che va digitata sulla tastiera come si scriverebbe sulla carta. La formula va scritta tenendo presenti alcuni punti:

Descrizione del programma

Linee
10-90

100-200

210

220-230

240-250

260-270

Commento

simboli atomici dei primi 102 elementi conosciuti, contenuti in istruzioni DATA

pesi atomici dei primi 102 elementi, contenuti in istruzioni DATA

dimensionamento delle matrici destinate a contenere i simboli ed i pesi atomici

memorizzazione dei dati nelle matrici: i dati relativi allo stesso atomo vengono messi in a\$ (simbolo) ed in a (peso atomico) in posizioni aventi lo stesso indice

INPUT formula

scomposizione della formula in caratteri separati; sistemazione dei singoli caratteri in vettore alfanumerico; inizializzazione variabili

280	inizio del loop di lettura della formula
290	controllo dell'eventuale presenza di caratteri estranei
300-320	ricostruzione dei simboli atomici
330	riconoscimento della situazione di parentesi aperta
340	riconoscimento della situazione di parentesi chiusa; ricostruzione del coefficiente moltiplicativo messo dopo la parentesi
350	calcolo del peso molecolare degli elementi contenuti all'interno della parentesi
360	azzeramento del flag di parentesi chiusa
370-380	riconoscimento del simbolo atomico; attribuzione del peso atomico corrispondente alla variabile x; riconoscimento di errori di battitura
390	ricostruzione dell'eventuale coefficiente moltiplicativo relativo ad un singolo atomo e moltiplicazione per il peso atomico
400-410	a seconda dello stato del flag di parentesi, il peso atomico di un elemento viene sommato alla variabile p (sommatoria all'interno parentesi) o alla variabile s (sommatoria totale)
420	chiusura del loop di lettura della formula
430	segnalazione di eventuali errori di battitura
440-450	visualizzazione del peso molecolare e ritorno alla fase di INPUT

Descrizione delle variabili

a\$(102)	simboli atomici
a(102)	pesi atomici
l\$(1+2)	formula da calcolare
p	flag di parentesi aperta (on) o parentesi chiusa (off)
b	simboli atomici sotto analisi
f	flag indicante l'azione di chiusura parentesi
l	lunghezza della formula
m,n	contatori
p	sommatoria dei pesi atomici all'interno delle parentesi
q	codice ASCII del carattere sotto analisi
s	sommatoria del peso molecolare
t	coefficiente moltiplicativo per atomi singoli
w	flag indicante stato di errore
x	peso atomico dell'elemento sotto analisi
y	coefficiente moltiplicativo dopo parentesi

1) Usare un solo livello di parentesi; per esempio, anziché scrivere $[Mg(OH)2]2$, usare la notazione $[Mg(OH)2]_2$. È possibile invece scrivere formule con più parentesi separate, come $(NH_4)_2Zn(SO_4)_2$.

2) Evitare di usare parentesi inutili, perché dopo una parentesi chiusa è obbligatorio mettere il coefficiente: per esempio, anziché $Na(OH)$ bisogna scrivere $Na(OH)_1$ oppure, più semplicemente, $NaOH$.

3) Nel caso si voglia ottenere il peso molecolare di un composto idrato, invece di usare la notazione classica $CuSO_4 \cdot 5H_2O$, scrivere $CuSO_4(H_2O)_5$.

4) Usare coefficienti con un massimo di due cifre.

5) Nel programma sono memorizzati, oltre ai 92 elementi naturali, i primi 10 elementi artificiali; è necessario evitare di inserire formule che comprendono atomi non memorizzati e che perciò risultano "sconosciuti" allo Spectrum.

In ogni caso, tutti gli eventuali errori di battitura vengono segnalati, evitando blocchi del programma.

Questa routine può essere utile per chi, studente o analista, vuole sviluppare programmi per la risoluzione di problemi di stechiometria; oppure può essere inserita in programmi didattici di chimica, mineralogia, eccetera.

I simboli ed i pesi atomici degli elementi riportati nel programma sono tratti da: Tabelle Logaritmiche per Chimici, Farmacisti, Medici e Fisici, di Küster e Thiel, 9ª edizione italiana (Hoepli).

Pesi molecolari

versione per ZX Spectrum 16K/48K

10	DATA "Ac", "Ag", "Al", "Am", "Ar", "As", "At", "Au", "B", "Ba", "Be"	110	DATA 9.013, 209, 249, 79.916, 12.011, 40.08, 112.41, 140.13, 251
20	DATA "Bi", "Bk", "Br", "C", "Ca", "Cd", "Ce", "Cf", "Cl", "Cm", "Co"	120	DATA 35.457, 247, 58.94, 52.01, 132.91, 63.54, 162.51, 167.27, 254
30	DATA "Cr", "Cs", "Cu", "Dy", "Er", "Es", "Eu", "F", "Fe", "Fm", "Fr"	130	DATA 152, 19, 55.85, 253, 223, 69.72, 157.26, 72.6, 1.008, 4.003
40	DATA "Ga", "Gd", "Ge", "H", "He", "Hf", "Hg", "Ho", "In", "Ir", "J"	140	DATA 178.5, 200.61, 164.94, 114.82, 192.2, 126.91, 39.1, 83.8
50	DATA "K", "Kr", "La", "Li", "Lu", "Md", "Mg", "Mn", "Mo", "N", "Na"	150	DATA 138.92, 6.94, 174.99, 256, 24.3, 2, 54.94, 95.95, 14.008, 22.991
60	DATA "Nb", "Nd", "Ne", "Ni", "No", "Np", "O", "Os", "P", "Pa", "Pb"	160	DATA 92.91, 144.27, 20.183, 58.71, 2, 53, 237, 16, 190.2, 30.975, 231
70	DATA "Pd", "Pm", "Po", "Pr", "Pt", "Pu", "Ra", "Rb", "Re", "Rh", "Rn"	170	DATA 207.21, 106.4, 147, 210, 140.92, 195.09, 242, 226.05, 85.48
80	DATA "Ru", "S", "Sb", "Sc", "Se", "Si", "Sm", "Sn", "Sr", "Ta", "Tb"	180	DATA 186.22, 102.91, 222, 101.1, 32.066, 121.76, 44.96, 78.96
90	DATA "Tc", "Te", "Th", "Ti", "Tl", "Tu", "U", "V", "W", "Xe", "Y", "Yb", "Zn", "Zr"	190	DATA 28.09, 150.35, 118.7, 87.63, 18, 0.95, 158.93, 99, 127.61
100	DATA 227, 107.88, 26.98, 243, 39.944, 74.91, 210, 197, 10.82, 137.36	200	DATA 232.05, 47.9, 204.39, 168.94, 2, 38.07, 50.95, 183.86, 131.3, 88.92, 1, 73.04, 65.38, 91.22

LOAD

Pesi molecolari

```
210 DIM a$(102,2): DIM a(102)
220 FOR n=1 TO 102: READ a$(n):
    NEXT n
230 FOR n=1 TO 102: READ a(n):
    NEXT n
240 CLS : INPUT "FORMULA : ";d$
250 PRINT AT 3,3;"FORMULA : ";d$
260 LET l=LEN d$: DIM l$(l+2)
270 FOR n=1 TO l: LET l$(n)=d$(n
    TO n): NEXT n: LET f=0: LET w=0
    : LET b$="": LET p$="off": LET s
    =0: LET t=1: LET x=1: LET p=0
280 FOR n=1 TO l
290 LET q=CODE l$(n): IF q<40 OR q>4
    1 AND q<48 OR q>57 AND q<65 OR q
    >90 AND q<97 OR q>122 THEN
    LET n=1: CLS : PRINT BRIGHT 1;
    FLASH 1;CHR$ q;" : SIMBOLO ESTR
    ANEO": PAUSE 150: GO TO 240
300 IF q>64 AND q<91 THEN LET b$=
    CHR$ q
310 IF CODE l$(n+1)>96 AND CODE l$(n
    +1)<123 THEN LET b$=b$+l$(n+1):
    LET n=n+1
320 IF LEN b$=1 THEN LET b$=b$+" "
330 IF l$(n)="(" THEN LET p=0:
    LET p$="on": GO TO 420
340 IF l$(n)=")" THEN LET f=1:
    LET y=VAL l$(n+1): IF CODE l$(n
    +2)>47 AND CODE l$(n+2)<58
    THEN LET y=y*10+VAL l$(n+2)
350 IF f=1 THEN LET p=p*y: LET b$="
    ": LET p$="off": LET n=n+1: IF y
    >9 THEN LET n=n+1
360 IF f=1 THEN LET x=0: LET f=0:
    GO TO 400
370 LET x=0: FOR m=1 TO 102: IF a$(m
    )=b$ THEN LET x=a(m): LET m=102
380 NEXT m: IF x=0 THEN LET w=1
390 IF CODE l$(n+1)>47 AND CODE l$(n
    +1)<58 THEN LET t=VAL l$(n+1):
    LET n=n+1: IF CODE l$(n+1)>47
    AND CODE l$(n+1)<58 THEN LET t
    =t*10+VAL l$(n+1): LET n=n+1
400 IF p$="off" THEN LET s=s+x*t+p:
    LET t=1: LET p=0: LET b$=""
410 IF p$="on" THEN LET p=p+x*t:
    LET t=1
420 NEXT n
430 IF w THEN CLS : PRINT BRIGHT 1
    ; FLASH 1;AT 10,7;"FORMULA SBAGL
    IATA": PAUSE 150: GO TO 240
440 PRINT AT 6,6;"P.M. : ";s
450 PAUSE 0: GO TO 240
```



Dalla grande edicola Jackson

Tutto sul personal computer

Personal

L'unica rivista indipendente per gli utenti dei personal computer Olivetti.
10 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 35.000

PERSONAL SOFTWARE

Aspetti e problemi del software per personal computer, programmi, giochi e sistemi operativi.
11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 34.000

Bit

La prima rivista europea di personal computer, software e accessori. Con test, novità, analisi del mercato...
11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 43.000

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana.
9 numeri all'anno: L. 2.000 a numero
Abbonamento: solo L. 15.000

PC

L'unica rivista italiana dedicata ai sistemi MS-DOS, Personal computer IBM e compatibili.
10 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 40.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:

Gruppo Editoriale Jackson via Rosellini, 12 - 20124 Mi

Desidero ricevere GRATIS un numero della Rivista

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

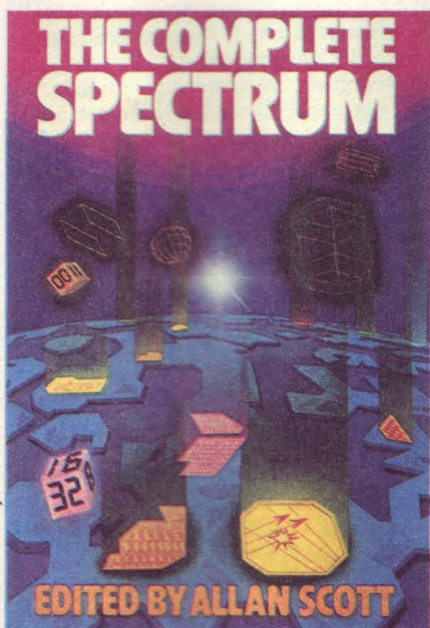
CAP _____ Città _____

Recensioni Libri

di **Lucio Bragagnolo**

The complete Spectrum (Lo Spectrum completo)

a cura di **Allan Scott**
ed. **Granada Publishing**



Ci è capitato finora di recensire libri sullo Spectrum dedicati ad argomenti specifici o di impostazione prettamente manualistica. Con "The complete Spectrum" ci troviamo invece ad affrontare per la prima volta una sorta di "opera omnia", che potrebbe benissimo intitolarsi "Tutto quello che vorreste sapere sullo Spectrum ecc.". Nonostante la mole gigantesca del libro (ben 482 pagine) il prezzo è insolitamente basso, tanto che viene da chiedersi se la Granada Publishing giochi al ribasso, abbia vinto una lotteria o ci abbia truffati fin dall'inizio della nostra attività. La risposta all'interrogativo è contenuta nel libro, che in effetti non è un'opera unica, bensì un collage ottenuto riunendo i capitoli più interessanti dei vari libri di

questa casa editrice dedicati allo Spectrum. Può essere interessante vederne insieme i titoli: "The ZX Spectrum and to get the most of it", "Spectrum graphics and sound", "40 educational games for the Spectrum", "The Spectrum book of games", "Introducing Spectrum machine code", "The Spectrum add-on guide", con autori quali Ian Sinclair, Allan Scott, Steve Money, Vince Apps, Mike James, Kay Ewbank e S. M. Gee. Vedete bene che non si tratta di illustri sconosciuti, anzi: alcuni di questi libri sono già apparsi sulle pagine di PRINT, a garanzia del loro valore e della loro affidabilità.

Più che un libro, insomma, si tratta di una compilation, in senso positivo, però: vale a dire che non si sta parlando di un'accozzaglia di contributi sparsi, ma, come dovrebbe accadere in teoria anche in campo discografico, una raccolta del meglio a disposizione. Ne è risultato un prodotto diretto, secondo quanto espresso nella prefazione, a tutta la famiglia, dove chiunque (esperto, principiante o giocherellone che sia) può trovare qualcosa di interessante.

Data la grande quantità di materiale presente nel libro, dovremo accontentarci di un esame sommario, passando velocemente in rassegna le varie parti di cui esso si compone. Anticipiamo fin d'ora che il viaggio sarà di lunghezza normale, ma toccherà proprio tutte le fermate, dall'apertura della confezione in cui viene venduto lo Spectrum fino al linguaggio macchina più ermetico. Trattenete il fiato ... si parte.

Il libro è diviso in parti (in totale 7) e capitoli (36), più quattro appendici situate alla fine del volume. La parte 1 ("Getting started") si occupa della messa in funzione del magico ZX, partendo dall'installazione della...spina sul cavetto di alimentazione (per chi non lo sapesse, i computer escono dagli stabilimenti Sinclair senza spina, lacuna che normalmente i vari importatori provvedono a colmare, adeguandosi ai vari standard esistenti) fino alle prime esperienze programmatiche.

Proseguendo, si arriva alla seconda parte, "Graphics and sound", dove il lettore viene introdotto al migliore utilizzo delle capacità grafiche dello Spectrum e dove compaiono alcuni programmi semplici, ma di effetto, consentendo cose come la rotazione e l'ingrandimento di figure, nonché il classico programma tipo "sketchpad" per disegnare sul video. La terza parte, "The leisure section", è dedicata ai giochi, educativi e no. La maggior parte del materiale presente proviene dai "40 educational games for the Spectrum".

Una delle sezioni più belle del libro è la numero 4, intitolata "Adventures with the Spectrum", i cui quattro capitoli sono dedicati al tema e, oltre a fornire il "solito" listato, presentano anche considerazioni e consigli chiari e utilissimi. Segue "Extending the Spectrum", insolita, ma notevole rassegna delle più tipiche espansioni per lo Spectrum, come joystick, tastiere aggiuntive, disk drive, sintetizzatori vocali, stampanti, penne ottiche, modem e chi più ne ha più ne interfacci. Il linguaggio macchina arriva alla parte 6, appunto "Introducing machine code", e viene trattato con buona chiarezza, senza però discostarsi molto dalla media degli altri libri. L'ultima sezione è intitolata "Appendices", ed è piuttosto atipica. Oltre a comprendere la solita lista di tutti i codici mnemonici Z80, infatti, vi sono altre tre appendici; una contenente la "lista selezionata del software Spectrum" (!!!), una dedicata al mercato delle espansioni e la terza riguardante gli...ordini per posta, forma di acquisto popolarissima in Inghilterra, dove peraltro pare che i ritardi siano l'eccezione e non la regola.

Parlando dell'aspetto generale del libro, la Granada ha creduto opportuno non rielaborare i vari testi per uniformarli, ottenendo sicuramente un risparmio di tempo e di denaro, a detrimento dell'estetica generale. Le varie parti sono così composte a volte in caratteri diversi, con listati stile ZX Printer contrapposti a quelli di stampanti professionali. In definitiva le varie parti del libro sono valide, se prese a sé stanti, mentre ne risultano evidenti tutte le diversità - anche di stile - se confrontate l'una con l'altra, conferendo all'opera un aspetto di patchwork che è sicura-

mente il suo difetto principale. Per chi bada alla sostanza, comunque, segnaliamo con piacere questa "summa" gigantesca, degna di occupare nella vostra biblioteca il posto d'onore, vicino al manuale dello Spectrum, e che saprà ripararvi e divertirvi in mille modi diversi. Controindicazioni? "The complete Spectrum" non è il libro fatto per chi è alla ricerca di dati approfonditi, mappe di memoria, disassemblati ROM e via dicendo. In più esso è scritto in lingua originale; come al solito con un linguaggio semplice e, a tratti, perfino spiritoso, ma sempre inglese. Tenetene conto...

L'hardware dello Spectrum - i componenti del Sinclair Spectrum e come intervenire nella sua struttura

di **Adrian C. Dickens**
ed. **Franco Muzzio & C.**



Quante volte abbiamo sentito parlare delle - a parole - infinite possibilità di interfacciamento del nostro piccolo computer con stampanti, frigoriferi, floppy disk drive, trenini elettrici, joystick, termostati, penne ottiche, robot e via dicendo? Sicuramente almeno qualcuna. I patiti del joystick come strumento di distruzione sistematica di alieni avranno fatto spallucce, girandosi di nuovo verso lo schermo per completare la trecentesima ondata del loro "Pacvaders" preferito. I super inesperti saranno stati presi dall'incredulità e, ostentando una finta indifferenza, avranno borbottato un accuratamente banale "Ah sì? Beh, può darsi ...", ripromettendosi segretamente di imparare in fretta

tutte quelle strane cose sul BASIC recitate dal loro manuale. Altri ancora, di età leggermente maggiore della media e in possesso di qualche conoscenza di elettronica, avranno, come si dice, rizzato le antenne, cominciando a intravedere nella loro scatoletta nera qualcosa di ben più attraente dei giochini e del software, simboleggiato dal connettore posteriore, prima ignorato e ora scrutato sospettosamente, quasi nascondesse chissà quali segreti. Alla fine, l'idea: proviamo a costruire qualcosa da interfacciare al computer!

E qui comincia il dramma. Il connettore posteriore si rivela essere un mistero ancora più grande del come utilizzarlo via software; mancano schemi elettrici, dati su cui si possa fare affidamento, elenchi dei componenti, e l'ignoranza della struttura interna spesso è totale. I più fortunati finiscono così per realizzare ugualmente qualcosa, grazie all'aiuto di qualche amico compiacente e competente; la maggioranza si ritira delusa, mentre i più sfortunati contemplanò i resti del proprio Spectrum ormai inservibili, anche se caldi e fumanti.

A chi è ancora in tempo per evitare una cruenta conclusione della propria carriera informatica consigliamo questo "L'hardware dello Spectrum", che diventerà sicuramente uno dei compagni fidati di chi ama pasticciare con diodi, resistenze, integrati e computer. Il libro, sebbene piccolo, è completissimo, comprendendo, oltre alle spiegazioni teoriche, tutto quanto serve in termini di schemi elettrici e fotografie, fornendo tutta la conoscenza teorica necessaria per disporre a piacimento dell'hardware Spectrum, magari per collegarvi addirittura una propria ROM o un altro microprocessore, naturalmente posto che si sappia poi programmarlo...

Nel corso dei 18 capitoli in cui si articola l'opera si potrà conoscere tutto di tutti i componenti dello Spectrum: alimentazione (3), CPU (4), memoria video (5), ROM (6), tastiera (7), ULA (8), espansione di memoria (9), circuito video (10) e, ovviamente, connettore posteriore (12), sia per quanto riguarda la vecchia (mai arrivata in Italia) Issue 1 che per la Issue 2 (la versione 3 non subisce cambiamenti sostanziali, così come quelle successive; l'ultima da noi vista era una Issue 6A, ricordate? Era quella dello Spectrum Plus da noi provata). La presenza di qualche pagina dedicata alla vecchia versione 1 ha un valore più che altro storico, ma può aiutare a capire la motivazione di scelte successive, nonché fornire alcune interessanti curiosità sulla genesi dei computer di sir Clive (il cui staff tecnico, scoprirete, ha avuto qualche problema non solo col QL), come lo "scarafaggio morto", un componente abbastanza insolito presente nella versione 1 a causa di una partita di ROM difettose. Gli altri capitoli sono abbastanza specifici, dedicati ad argomenti come la diagnosi dei qua-

sti, come aggiungere un PIO Z80A, come collegare una tastiera supplementare autocostruita o un joystick, e altro. Nel corso del libro verranno chiariti tanti piccoli misteri, spesso tali anche per un non ignorante di elettronica: perché lo Spectrum riscalda così tanto? Quanta corrente è possibile prelevare dall'alimentatore? Quali sono i difetti hardware dello Spectrum? Perché gli Spectrum 16K presentano irregolarità nel funzionamento di programmi di temporizzazione? È possibile variare il clock dello Z80? Molta carne al fuoco, quindi: per di più tutta "arrostita" con un linguaggio chiaro e da schemi ben fatti e più esplicitivi di mille parole. Si vedrà anche come procedere in concreto, attraverso l'illustrazione delle tecniche di "breadboarding" e "veroboarding", alla realizzazione di alcuni dispositivi, con immagini fotografiche e schemi di ciò che viene costruito. Al termine alcune appendici, dedicate a riferimenti bibliografici, chiarimenti sui termini usati nel testo, un elenco dei componenti dello ZX Spectrum e il relativo schema elettrico. Come ciliegina finale sulla torta, nel libro sono contenuti anche gli indirizzi a cui rivolgersi per reperire le basette necessarie a chi si vuol cimentare nell'arte del veroboarding.

In definitiva, un testo insostituibile per chi s'intende di hardware Spectrum ... o vorrebbe cominciare ad intendersene. Già che ci siamo, prima di salutarvi pubblichiamo un assaggio del libro, più precisamente un listatino che visualizza sul video un problema hardware dello Spectrum. Provatelo e ... incuriositevi.

```
10 CLEAR 32499
15 PRINT "inserisci valori compresi
tra 64 e 127"
20 INPUT "valore del registro I ? ";v
30 POKE 32500,62: REM ld a,v
40 POKE 32501,v
50 POKE 32502,237: REM ld i,a
60 POKE 32503,71
70 POKE 32504,201 : REM ret
80 LET a=USR 32500
90 GO TO 20
```

**I libri italiani
vengono offerti
con il 10% di
sconto sul
prezzo di copertina**

**I libri possono
essere ordinati
utilizzando il
coupon pubblicato
a pag. 82**

ZX CLUB

I club
Sinclair

Gli annunci di ZX Club

Il Gruppo Utilizzatori Computer Sinclair Napoli organizza per l'estate prossima una favolosa gita con visita guidata alla Sinclair Research di Cambridge. La gita, della durata di una settimana (compreso il viaggio A/R in aereo), è riservata ai possessori di computer Sinclair che abbiano superato il 18° anno di età o, se minorenni, accompagnati da persona responsabile. La quota di partecipazione risulta essere molto conveniente, in quanto il discreto numero di partecipanti previsti consente di usufruire di sconti molto particolari. Per maggiori informazioni scrivere o telefonare alla segreteria del gruppo, tenendo presente che una eventuale prenotazione deve essere effettuata entro il 10 maggio. Affrettati: non è questa l'occasione che aspettavi per visitare l'Inghilterra?

L'indirizzo del Club è:

GRUPPO UTILIZZATORI COMPUTER SINCLAIR
c/o Roberto CHIMENTI
Via Luigi Rizzo, 18
80124 Napoli NA
Tel.: 081/7623121

Un giovane amico polacco desidera ricevere listati da 1K su carta per lo ZX81.

Siamo certi che i lettori vorranno aiutare questo amico che, evidentemente, si trova in difficoltà a causa della scarsa diffusione dell'informatica (e dell'informazione) nella sua terra.

L'indirizzo è:

JOZEF PIOTR MROWIEC
ul. Anlota nr 4
Skr. poczt. 2205
40-856 Katowice-Zaleze
POLONIA

Per iniziativa di un gruppo di lettori di Sesto San Giovanni è nato il QL USER CLUB! Il Q.U.C. è un club senza fini di lucro, nato per favorire i contatti e gli scambi di idee e di materiale hard-soft-didattico fra i suoi soci.

L'iscrizione è gratuita e non comporta alcun tipo di impegno. Fra gli scopi del club vanno particolarmente segnalati la intenzione di costituire una biblioteca (libri, riviste, software) a disposizione dei soci, l'intenzione di pubblicare un bollettino periodico, la volontà di interessarsi anche degli aspetti hardware del QL, così da agevolare chi volesse realizzare prototipi di accessori interessanti. Chi desidera mettersi in contatto con il club può rivolgersi a:

QL USER CLUB
c/o Roberto GHEZZI
Via Volontari del Sangue, 202
20099 Sesto San Giovanni MI

I programmi di ZX Club

Questo programma è una raccolta di routine, che possono essere facilmente estrapolate ed inserite nei vostri programmi. Tutte le routine sono rilocabili: potrete provarle una ad una e, eventualmente, salvarle su nastro sotto forma di byte. Il programma prevede anche una opzione di Demo, sfruttando la quale potrete rapidamente farvi una

idea dei risultati conseguibili grazie a queste routine. L'analisi del listato è facilitata dalla semplicità della parte BASIC e dalle REM opportunamente posizionate.

E' la linea 80 che effettua il salto alla routine richiesta dall'utente, attraverso la selezione del suo numero d'ordine dal menu principale. La linea 80 può sembrare complessa, ma si è preferita tale soluzione al posto di una lunga serie di IF...THEN GO TO che avrebbe reso il listato fortemente involuto. Dalla linea 105 in poi troviamo le linee di DATA in cui sono collocati tutti i numeri decimali relativi alle routine in oggetto. Notate che i numeri sono stati inseriti come stringhe, allo scopo di economizzare memoria. Alla linea 7470 troviamo il sottoprogramma caricatore, che ci permette di collocare in memoria la routine richiesta, e alla linea 7512 quello che ci permette di sperimentarla.

Chi possiede lo Spectrum 16K potrà visionare e sfruttare le varie routine, caricandole in memoria ad una ad una ed adattando opportunamente i programmi caricatore ed esecutore.

**leggete
papersoft**

**Raccolta Routine di Antonio
Danese
Gruppo Utilizzatori Computer
Sinclair Napoli
Versione per ZX Spectrum
48K**

```

1 REM Questa linea deve contenere
  almeno tre righe di caratteri al
  trimenti il l/m sporcherà il pr
  ogramma
2 REM
  Raccolta Routine l/m
                                     di
  Danese Antonio                       Via
  Cavallegeri Aosta 35                 Nap
  oli
3 REM                                     Grupp
  o Utilizzatori Computer              Sincl
  air Napoli -1985-
10 INK 9: RESTORE 8000: FOR a=1
  TO 18 STEP 17: CLS : PRINT AT 0
  ,9;"Routine l/m": PRINT AT 3,0;:
  BEEP .2,14: BEEP .2,16: BEEP .4
  ,18: PRINT
20 FOR b=a TO (a+16): READ c$:
  PRINT b;" ";c$: BEEP .009,50:
  NEXT b: BEEP .5,0: BEEP .5,2
30 PRINT #0; PAPER 4; INK 0;" ";
  FLASH 1;"C"; FLASH 0;"ontinua.
  "; FLASH 1;"D"; FLASH 0;"emo. ";
  FLASH 1;"N"; FLASH 0;"umero.
  "
40 PAUSE 0
50 LET a$=INKEY$: GO TO VAL "40"+(a
  $="c")*VAL "20"+(a$="d")*VAL "84
  60"+(a$="n")*VAL "35"
60 NEXT a: GO TO VAL "10"
70 GO TO VAL "10"
75 INPUT "Quale numero vuoi? ";
  LINE a$
80 GO TO 3+(a$="1")*100+(a$="2")*20
  0+(a$="3")*300+(a$="4")*400+(a$=
  "5")*500+(a$="6")*600+(a$="7")*7
  00+(a$="8")*800+(a$="9")*900+(a$
  ="10")*1000+(a$="11")*1100+(a$="
  12")*1200+(a$="13")*1300+(a$="14
  ")*1400+(a$="15")*1500+(a$="16")
  *1600+(a$="17")*1700+(a$="18")*1
  800+(a$="19")*1900+(a$="20")*200
  0+(a$="21")*2100+(a$="22")*2200+
  (a$="23")*2300+(a$="24")*2400+(a
  $="25")*2500+(a$="26")*2600+(a$=
  "27")*2700+(a$="28")*2800+(a$="2
  9")*2900+(a$="30")*3000+(a$="31"
  )*3100+(a$="32")*3200+(a$="33")*
  3300+(a$="34")*3400+(a$="35")*35
  00+(a$="36")*3600
105 CLS : PRINT "Routine n 1
  Scroll 8 pixel a si
  nistra": RESTORE 105: GO TO 7450
150 DATA "34","6","8","33","31","64"
  ,"22","0","30","65","14","32","1
  75","126","23","119","43","13","

```

```

32","249","25","62","88","188","
40","3","43","24","235","5","175
  ","184","32","225","201"
205 CLS : PRINT "Routine n 2
  Scroll 8 pixel a de
  stra": RESTORE 205: GO TO 7450
250 DATA "26","6","8","33","0","64",
  "14","32","175","126","31","119"
  ,"35","13","32","249","62","88",
  "188","32","241","5","175","184"
  ,"32","233","201"
305 CLS : PRINT "Routine n 3
  Scroll 1 pixel a si
  nistra": RESTORE 305: GO TO 7450
350 DATA "26","33","255","87","6","1
  92","197","229","221","225","221
  ","126","225","7","55","63","6",
  "32","203","22","43","16","251",
  "193","16","236","201"
405 CLS : PRINT "Routine n 4
  Scroll 1 pixel a si
  nistra e ritorno":
  RESTORE 405: GO TO 7450
450 DATA "26","33","255","87","6","1
  92","197","229","221","225","221
  ","126","225","7","63","63","6",
  "32","203","22","43","16","251",
  "193","16","236","201"
505 CLS : PRINT "Routine n 5
  Scroll 1 pixel a si
  n e ritorno ed inverse":
  RESTORE 505: GO TO 7450
550 DATA "26","33","255","87","6","1
  92","197","229","221","225","221
  ","126","225","7","0","63","6","
  32","203","22","43","16","251",
  "193","16","236","201"
605 CLS : PRINT "Routine n 6
  Scroll 1 pixel a De
  stra": RESTORE 605: GO TO 7450
650 DATA "26","33","0","64","6","192
  ","197","229","221","225","221",
  "126","31","15","55","63","6","3
  2","203","30","35","16","251","1
  93","16","236","201"
705 CLS : PRINT "Routine n 7
  Scroll 1 pixel a De
  stra e ritorno":
  RESTORE 705: GO TO 7450
750 DATA "26","33","0","64","6","192
  ","197","229","221","225","221",
  "126","31","15","63","63","6","3
  2","203","30","35","16","251","1
  93","16","236","201"
805 CLS : PRINT "Routine n 8
  Scroll 1 pixel a De
  stra ritorno ed inverse"
  : RESTORE 805: GO TO 7450
850 DATA "26","33","0","64","6","192
  ","197","229","221","225","221",
  "126","31","15","0","63","6","32
  ","203","30","35","16","251","19
  3","16","236","201"

```

```

905 CLS : PRINT "Routine n 9
          Scroll 1 pixel in a
          lto": RESTORE 905: GO TO 7450

950 DATA "55","33","31","64","84","9
3","37","6","3","197","6","8","1
97","6","8","197","36","1","32",
"0","237","184","17","32","0","2
5","84","93","193","16","240","1
","32","248","9","193","16","230
","62","7","132","103","193","16
","220","33","223","87","6","32"
,"35","54","0","16","251","201"

1005 CLS : PRINT "Routine n 10
          Scroll 1 pixel in b
          asso": RESTORE 1005: GO TO 7450

1050 DATA "53","33","255","87","84","
93","36","6","3","197","6","8","
197","6","8","197","37","1","32"
,"0","237","184","17","32","0","
25","84","93","193","16","240","
1","224","7","9","193","16","230
","62","249","132","103","193","
16","220","37","6","32","35","54
","0","16","251","201"

1105 CLS : PRINT "Routine n 11
          Scroll 1 pixel in a
          lto e ritorno": RESTORE 1105:
          GO TO 7450

1150 DATA "57","33","0","64","17","20
8","92","1","32","0","237","176"
,"6","191","33","0","64","197","
93","84","36","124","230","7","3
2","10","125","198","32","111","
56","4","124","214","8","103","2
29","1","32","0","237","176","22
5","193","16","227","17","224","
87","33","208","92","1","32","0"
,"237","176","201"

1205 CLS : PRINT "Routine n 12
          Scroll 1 pixel in b
          asso
          e ritorno":
          RESTORE 1205: GO TO 7450

1250 DATA "59","33","224","87","17","
208","92","1","32","0","237","17
6","6","191","33","224","87","19
7","93","84","37","124","230","7
","254","7","32","10","125","214
","32","111","56","4","124","198
","8","103","229","1","32","0","
237","176","225","193","16","225
","17","0","64","33","208","92",
"1","32","0","237","176","201"

1305 CLS : PRINT "Routine n 13
          Scroll superiore a
          destra
          e ritorno":
          RESTORE 1305: GO TO 7450

1350 DATA "25","6","64","17","255","7
1","213","225","43","197","1","3
1","0","26","237","184","35","11
9","0","43","43","27","193","16"
,"240","201"

1405 CLS : PRINT "Routine n 14
          Scroll centrale a d
          estra
          e ritorno":
          RESTORE 1405: GO TO 7450

1450 DATA "25","6","64","17","255","7
9","213","225","43","197","1","3
1","0","26","237","184","35","11
9","0","43","43","27","193","16"
,"240","201"

1505 CLS : PRINT "Routine n 15
          Scroll inferiore a
          destra
          e ritorno":
          RESTORE 1505: GO TO 7450

1550 DATA "25","6","64","17","255","8
7","213","225","43","197","1","3
1","0","26","237","184","35","11
9","0","43","43","27","193","16"
,"240","201"

1605 CLS : PRINT "Routine n 16
          Scroll superiore e
          centrale
          a destra e ritorno"
          : RESTORE 1605: GO TO 7450

1650 DATA "25","6","128","17","255","
79","213","225","43","197","1","
31","0","26","237","184","35","1
19","0","43","43","27","193","16
","240","201"

1705 CLS : PRINT "Routine n 17
          Scroll inferiore e
          centrale
          a destra e ritorno"
          : RESTORE 1705: GO TO 7450

1750 DATA "25","6","128","17","255","
87","213","225","43","197","1","
31","0","26","237","184","35","1
19","0","43","43","27","193","16
","240","201"

1805 CLS : PRINT "Routine n 18
          Scroll della parte
          medio bassa verso l'alto":
          RESTORE 1805: GO TO 7450

1850 DATA "6","6","17","205","0","14"
,"201"

1905 CLS : PRINT "Routine n 19
          Scroll attributi e
          ritorno": RESTORE 1905: GO TO 74
          50

1950 DATA "52","62","255","1","0","0"
,"17","32","24","33","223","87",
"245","213","17","32","0","4","2
5","16","253","65","4","35","16"
,"253","209","66","241","197","6
","0","75","13","213","229","209
","26","213","35","237","176","1
8","225","17","32","0","25","209
","193","16","233","201"

2005 CLS : PRINT "Routine n 20
          Scroll a sinistra c
          on scomparsa": RESTORE 2005:
          GO TO 7450

```

```

2050 DATA "18","33","255","87","14","
32","167","203","38","43","13","
32","250","62","63","188","32","
242","201"

2105 CLS : PRINT "Routine n 21
          Scroll a destra con
          scomparsa": RESTORE 2105:
          GO TO 7450
2150 DATA "18","33","0","64","14","32
","167","203","62","35","13","32
","250","62","88","188","32","24
2","201"
2205 CLS : PRINT "Routine n 22
          Scroll a sinistra c
          on con          ritorno nello stess
          o byte": RESTORE 2205: GO TO 745
          0
2250 DATA "18","33","255","87","14","
32","167","203","6","43","13","3
2","250","62","63","188","32","2
42","201"
2305 CLS : PRINT "Routine n 23
          Scroll a destra con
          con          ritorno nello stess
          o byte": RESTORE 2305: GO TO 745
          0
2350 DATA "18","33","0","64","14","32
","167","203","14","35","13","32
","250","62","88","188","32","24
2","201"
2405 CLS : PRINT "Routine n 24
          CLS a tendina":
          RESTORE 2405: GO TO 7450
2450 DATA "43","33","0","64","17","0"
,"24","54","255","35","27","123"
,"178","32","248","33","0","64"
,"17","0","24","54","0","35","27"
,"123","178","32","248","33","0"
,"88","17","192","2","54","15","
35","27","123","178","32","248"
,"201"

2505 CLS : PRINT "Routine n 25.
          CLS ": RESTORE 2505
          : GO TO 7450
2550 DATA "45","14","32","33","0","88
","6","24","17","32","0","229","
54","18","25","16","251","197","
33","208","0","17","32","0","205
","181","3","193","225","229","6
","24","17","32","0","54","9","2
5","16","251","225","35","13","3
2","217","201"

2605 CLS : PRINT "Routine n 26
          CLS parziale ":
          RESTORE 2605: GO TO 7450
2650 DATA "6","6","12","205","68","14
","201"

2705 CLS : PRINT "Routine n 27
          CLS e BORDER ":
          RESTORE 2705: GO TO 7450

2750 DATA "38","243","6","25","197","
33","0","64","22","0","62","236"
,"6","25","35","94","245","123"
,"211","254","241","43","115","35
","16","244","114","35","61","32
","237","193","16","226","205","
107","13","251","201"

2805 CLS : PRINT "Routine n 28
          Effetti BORDER 1 ":
          RESTORE 2805: GO TO 7450
2850 DATA "17","14","255","6","255","
120","211","254","16","251","13"
,"32","246","62","1","211","254"
,"201"
2905 CLS : PRINT "Routine n 29
          Effetti BORDER 2 ":
          RESTORE 2905: GO TO 7450
2950 DATA "16","243","6","6","197","6
","255","120","211","254","16","
251","193","16","245","251","201
"

3005 CLS : PRINT "Routine n 30
          Effetti BORDER 3 ":
          RESTORE 3005: GO TO 7450
3050 DATA "17","14","70","6","99","12
0","211","254","16","251","13","
32","246","62","6","211","254","
201"

3105 CLS : PRINT "Routine n 31
          Effetti PAPER":
          RESTORE 3105: GO TO 7450
3150 DATA "124","175","50","4","91","
62","31","50","0","91","62","23"
,"50","1","91","33","0","88","34
","2","91","6","12","197","42","
2","91","58","0","91","71","58"
,"4","91","119","35","16","252","
58","1","91","71","58","4","91"
,"17","32","0","119","25","16","2
52","42","2","91","58","1","91"
,"71","58","4","91","17","32","0"
,"119","25","16","252","58","0"
,"91","71","58","4","91","119","3
5","16","252"

3151 DATA "119","42","2","91","1","33
","0","9","34","2","91","33","0"
,"91","53","53","35","53","53","
38","1","58","4","91","111","17"
,"10","0","205","181","3","193"
,"16","165","58","4","91","254","
56","200","198","8","24","134","
201"

3205 CLS : PRINT "Routine n 32
          Scroll a scomparsa"
          : RESTORE 3205: GO TO 7450
3250 DATA "17","33","0","64","1","0"
,"24","203","6","203","134","35"
,"11","120","177","32","246","201
"

```

```

3305 CLS : PRINT "Routine n 33
          Trasferimento video
          alto": RESTORE 3305: GO TO 7450

3350 DATA "23","17","0","88","33","0"
          ,"152","1","0","1","237","176","
          17","0","64","33","0","128","1"
          ,"255","7","237","176","201"

3405 CLS : PRINT "Routine n 34
          Trasferimento video
          totale": GO SUB 7450: RESTORE 3
          405: GO TO 7470/

3450 DATA "12","17","0","64","33","0"
          ,"128","1","0","27","237","176"
          ,"201"

7450 PRINT AT 10,0;"A Quale indirizzo
          la vuoi inserire?
          (min: 40000)": INPUT indirizzo:
          IF indirizzo<40000 THEN
          GO TO 7450

7470 READ l$: LET lung=VAL l$: FOR n=
          indirizzo TO (indirizzo+lung-1):
          READ m$: LET lm=VAL m$: POKE n,
          lm: NEXT n

7500 PRINT AT 10,0;"Routine inserita
          in memoria
          ""CODE ";indiriz
          zo;",";lung;"
          ":
          PRINT AT 15,0;"Vuoi registrarla
          ?
          ": PAUSE 0: LET a$
          =INKEY$: GO TO 7500+(a$="s")*5+(
          a$="n")*10

7505 PRINT AT 15,0;"Quale nome le vuo
          i dare?
          ": INPUT LINE n$:
          GO TO 7505+(LEN n$>0 AND LEN n$
          <11)*1

7506 PRINT AT 15,0;"
          ": PRINT AT 10,0;n
          $;"
          ": SAVE n$CODE indirizzo
          ,lung: PRINT AT 15,0;"Vuoi verif
          icare la qualita' della regi
          strazione?
          "

7507 PAUSE 0: LET a$=INKEY$: GO TO 75
          07+(a$="s")*1+(a$="n")*3

7508 PRINT AT 15,0;"Fammi riascoltare
          la registrazione
          ": VERIFY n$CODE indiriz
          zo,lung: PRINT AT 15,0;"
          VERIFY O.K
          ": PAUSE 100

7510 PRINT AT 15,0;"Vuoi provarla?
          ": PAUSE 0: LET a$
          =INKEY$: GO TO 10+(a$="s")*7501

7511 CLS : FOR n=0 TO 21: PRINT
          PAPER 3;"GRUPPO"; PAPER 4;"UTIL

```

```

          IZZATORI"; PAPER 5;"COMPUTER";
          PAPER 6;"NAPOLI": NEXT n:
          PRINT #0;"Premi un tasto x iniz
          iare
          poi un'altro per il m
          enu'"

7512 PAUSE 0: RANDOMIZE USR indirizzo
          : FOR n=0 TO 10E10: IF INKEY$=""
          THEN RANDOMIZE USR indirizzo:
          NEXT n

7513 GO TO 10

8000 DATA " Scroll 8 pix a Sx"," Scro
          ll 8 pix a Dx"," Scroll 1 pix a
          Sx"," Scroll 1 pix a Sx e rit","
          Scroll 1 pix a Sx rit e inv"

8001 DATA " Scroll 1 pixel a Dx"," Sc
          roll 1 pixel a Dx e rit"," Scrol
          l 1 pix a Dx rit e inv"," Scroll
          1 px in alto","Scroll 1 pix in
          basso"

8002 DATA "Scroll 1 pix in alto e rit
          ","Scroll 1 pix in basso e rit",
          "Scroll sup a Dx e rit","Scroll
          cent a Dx e rit","Scroll inf a D
          x e rit"

8003 DATA "Scroll sup e cent a Dx e r
          it","Scroll inf e cent a Dx e ri
          t","Scroll med/bassa in alto","S
          croll ATTR e rit","Scroll a Sx c
          on scomparsa"

8004 DATA "Scroll a Dx con scomparsa"
          ,"Scroll a Sx con rit(=)Bytes",
          "Scroll a Dx con rit(=)Bytes","CL
          S a tendina","CLS da Sx a Dx"

8005 DATA "CLS parziale","CLS ed effe
          tti Bordo","Effetti BORDO 1",
          "Effetti BORDO 2","Effetti BORDO 3"
          ,"Effetti PAPER","CLS a scompars
          a","Trasferimento video alto",
          "T rasferimento video totale"

8500 CLS : PRINT AT 10,10; FLASH 1;"D
          IMOSTRAZIONE": FLASH 0: PRINT
          AT 15,0;"Ora ti verranno mostrat
          e tutte le routine presenti in
          questo programma.Per cambiare
          routine premi un tasto""Premi
          un tasto per iniziare": PAUSE 0
          : FOR n=100 TO 3300 STEP 100:
          RESTORE n: GO SUB 8600: NEXT n

8600 READ l$: LET lung=VAL l$: FOR b=
          61000 TO (61000+lung-1): READ m$
          : LET lm=VAL m$: POKE b,lm:
          NEXT b: BORDER 7: PAPER 7:
          INK 9: CLS : FOR m=0 TO 21:
          PRINT PAPER 3;"GRUPPO";
          PAPER 4;"UTILIZZATORI"; PAPER 5
          ;"COMPUTER"; PAPER 6;"NAPOLI":
          NEXT m: RANDOMIZE USR 61000:

```

```

PAUSE 10: RANDOMIZE USR 61000:
PAUSE 10: FOR b=0 TO 100000:
IF INKEY$="" THEN RANDOMIZE
USR 61000: NEXT b
8601 IF n<3400 THEN RETURN
8602 GO TO 10
9000 CLS : PRINT AT 10,2; FLASH 1;"F
E R M A I L N A S T R O";#0;
FLASH 0;" Premi un tasto per i
niziare"
9001 PAUSE 0
9002 CLS : PRINT INK 4;"{SG4}{30G3}
{G7}": FOR n=1 TO 20: PRINT
INK 4;"{SG5}";TAB 31;"{G5}":
NEXT n: PRINT INK 4;"{SG1}
{30SG3}{SG2}"
9004 PRINT AT 1,10;"Routine 1/m";AT 3

```

```

,2;"Questo programma vi consente
";AT 4,1;"di collocare a qualsia
si";AT 5,1;"indirizzo le routine
in esso";AT 6,1;"contenute e di
registrarle ";AT 7,1;"su nastro
. Le opzioni sono:";AT 9,1;"C.co
ntinua ~cambia pagina~";AT 11,1;
"D.demo ~vi consente di vedere";
AT 12,3;"tutte le routine presen
ti";AT 13,3;"nel programma~";
AT 15,1;"N.numero ~per scegliere
una";AT 16,3;"particolare routi
ne~";AT 20,16;"Antonio Danese"
9005 PRINT #0;" Premi un tasto per i
l menu'": PAUSE 0: CLS : GO TO 1
0
9995 STOP
9999 SAVE "Rout.1/m" LINE 9000

```

L. 2.000

Specialità in Abbonamento Postale - gruppo 11079

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana

VI RICORDATE DEL MAESTRO MANZI?

A parlare è Egidio Pentiraro, autore di: "A scuola con il computer" e "Computer è facile". Dopo aver insegnato per due anni

pedagogia informatica all'Università di Sassari si sta dedicando alla produzione di unità didattiche in Logo, e di editoria elettronica.

A chi punta il tallone scolastico con il computer?

In Italia non c'è una grossa produzione di unità didattiche, mentre esiste un certo numero di insegnanti in grado di scrivere un libro di testo, questo non accade per il software, il problema è che si devono creare autori di SW nella stessa proporzione in cui si sono creati autori di libri. In Inghilterra il 2% degli insegnanti è autore di libri scolastici.

In attesa del piano ministeriale cosa si può fare?

Abbiamo sempre chiesto un piano, che avrebbe potuto avere gradi di libertà, che avrebbe potuto prendere ad esempio il modello inglese o francese. Un piano, avrebbe dovuto riunire tutti quelli che si occupavano di informatica, ed ognuno avrebbe dovuto contribuire per la propria parte: scuole, editori, insegnanti, case produttrici di HW e di SW.

Ci sono stati in luogo del piano cassettoni, alcuni di buon livello altri molto meno. Il computer d'attorno è arrivato nella scuola, nella maggior parte dei casi solamente in senso fisico, e sull'onda dell'entusiasmo, di alcune persone. Poi naturalmente gli entusiasmi si sono spenti ed oggi la situazione comincia ad essere in alcune scuole, leggermente incrinata. Siamo in una situazione molto delicata ed anche molto pericolosa, gli insegnanti hanno capito che per portare a scuola il computer, non basta comperare una macchina, ma occorre portare tutto un apparato di conoscenze (segue a pag. 2)

6

Bacheca
Notizie e notizie, annunci e iniziative. (pag. 2)

TANTAM
Gli insegnanti, le scuole, gli studenti presentano il loro lavoro. (pag. 6)

Software
Abbiamo visionato per voi. (pag. 8)

Compete ovunque
Un itinerario per diventare esperti. (pag. 10)

Statistica per gli insegnanti
Parte seconda. (pag. 14)

Un video a sostegno
Diastole, scuola e computer. (pag. 19)

La Terra di Babele
Comunicare con il calcolatore - seconda parte. (pag. 20)

Microelettronica a scuola
La politica inglese. (pag. 26)

A.B.C.
Dizionario di informatica. (pag. 29)

DIDATTICA '85

Si è concluso il concorso Didattica '85. "Sudare con il Calcolatore". Ecco i vincitori:

Stefano Donati, SMS "G. Cardano", Gallarate (VA)
Silvano Marzani, SMS "G. Govi", S. Matteo delle Chiaviche (MN)
Gianluca Rotta, SMS "M. Craspi", Ghemme (NO)
Alessandro Chinelli, I.T.S. "E. Torricelli", Milano
Alberto Belloni e Maurizio Meneghino, Liceo Galileo "L. Pavesi", Longo (VI)

Umberto Molteni, Liceo Scientifico Statale di Zogno (BG)
Paolo Baroni, Liceo Scientifico Statale di Zogno (BG)
Franco Michele Tavella, I.T.S. Geometri "C. Castaneo", Milano
Claudia Salvati, SMS "G. Peracca", Pizzato (BS)

L. 2.000

IN EDICOLA

Una pubblicazione firmata...



GRUPPO EDITORIALE JACKSON

San Francisco - Londra - Milano

input.output

Piccoli
annunci

Cambio software in LM per Spectrum 48 Kbyte. Inviatemi il vostro elenco, vi invierò il mio di circa 300 programmi tra utility e giochi. Fulvio Galeotti - Viale Po, 58 - 26100 Cremona - Tel. 0372/410372

Cambio software per ZX Spectrum. Scelta fra oltre 500 programmi. Giudizio di validità per evitare bidoni. Raccolte per generi. Chiedere lista gratuita. Beppe Fasolis - Corso Alba, 13 - 14100 Asti - Tel. 0141/53817

Vendo per Spectrum 16 Kbyte cassette con minimo 15 giochi a L. 800 a gioco. Richiedere lista o telefonare a: Sciuto Nino - Via G. Leopardi, 74 - 95127 Catania - Tel. 091/386890

Vendo per ZX Spectrum programmi 16-48 Kbyte, rispettivamente a L. 2.000-3.000. Scrivere o telefonare a: Massimo Mondella - Via Ernesto Mauri, 12 - 00171 Roma - Tel. 06/255147

Vendo cassette giochi duplicate per Spectrum 16-48 Kbyte a L. 2.000. Per ricevere il catalogo allegare francobolli. Rispondo a tutti. Vi aspetto - Roberto Negri - Via Milano - 43039 Salsomaggiore (PR)

Vendo per Spectrum 48 Kbyte 18 giochi, tutti in linguaggio macchina, tra cui: Atic Atac, Jet Set Willy, Flight Simulation, ecc. a sole L. 30.000. Aldo Bergantin - Via Vipacco, 45 - 10142 Torino - Tel. 011/4110036

Vendo programmi per ZX Spectrum 16-48 Kbyte fra i migliori esistenti: Match Point, Kung Fu, Psytron, Sabre Wulf, Mugsy, ecc. a L. 3.000 l'uno. Massima serietà. Luca Cappellini - Via Marco Polo, 7 - 20124 Milano - Tel. 02/6571523

Vendo, cambio programmi per ZX Spectrum 16-48 Kbyte. Prezzi stracciati. Giuseppe Fortunato - Via Veneto, 1 - 70123 Bari - Tel. 080/371757

Vendo e cambio programmi per ZX Spectrum e C64 su nastro a prezzi insuperabili. Richiedere ed inviare la lista. Ugo Drogo - Via Cavour - 92024 Canicattì (AG)

Cambio o vendo programmi per Spectrum 16-48 Kbyte a prezzi fantastici (provare per credere). Inviatemi la vostra lista oppure telefonatemi. E' una vera occasione. Alessandro Cantù - Via Badia, 15 - 27049 Stradella (PV) - Tel. 0385/48432

Vendo programmi per ZX Spectrum 16 Kbyte a sole L. 1.500 e 48 Kbyte a L. 2.000 (1/10 del prezzo reale), fra cui: Jet Set, Hobbit, Olimpiadi, Match Point, Atic Atac, Sabre Wulf, Alchemist. Sconti per grossi quantitativi. Richiedere catalogo. Telefonare ore pastri, Emiliano Cecchetti - Viale Duca D'Aosta, 13 - 21052 Busto Arsizio (VA) - Tel. 0331/631664

Hello! Disponiamo di più di 400 programmi (in continuo aumento): video, Arcade, Adventure, games ed utilities con le ultime novità inglesi a L. 2.000 (16 Kbyte) e L. 3.000 (48 Kbyte). Attenzione: cassetta con 20 programmi a scelta al prezzo irrisorio di L. 25.000. Massimiliano Panarese - Via Dei Giardini, 6 - 00048 Nettuno (RM) - Tel. 9802297

Vendo programmi per ZX Spectrum a L. 5.000 l'uno, compresa cassetta e istruzioni originali. Disponibili ultime novità inglesi. Maurizio Leone - Via Gaio Melisso, 16 - 00175 Roma - Tel. 06/7662671

Vendo a L. 30.000 oltre 100 programmi tra cui i favolosi Jet Pac, Trans Am, Invaders. Per avere la lista software scrivete: non mancherò di rispondervi. Cosimo Laserra - Via M. L. King, 21 - 70124 Bari - Tel. 080/510564

Vendo, cambio software per Spectrum 16-48 Kbyte. Dispongo delle ultime novità inglesi in fatto di games, ma ho anche programmi di utilità. Telefonare o richiedere lista. Andrea Zavaglia - Via S. Pascucci, 28 - 47035 Gambettola (FO) - Tel. 53226

Vendo cassette per Spectrum 16 o 48 Kbyte a prezzi incredibili! Scrivetemi per una lista con i titoli e i prezzi. Es. Flight Simulation L. 13.000, Alchemist L. 8.000, ecc. Erik Bregoli - Via Trento, 18 - 20081 Abbiategrasso (MI) - Tel. 02/9461610

Straordinario! Vendo per ZX Spectrum 48 Kbyte ben 40 giochi a sole L. 20.000 più spese postali. Fra i giochi: Chequered Flag, Ulrich (con ben 37 schemi), Moon Patrol, World Cup (Calcio), programma per fare facilmente i videogiochi in LM e tanti altri. Vendo inoltre a L. 10.000 più spese postali cassetta con 20 giochi per ZX Spectrum 16 Kbyte fra cui: Explorer, Cokkie, Pstt, Repulsar e parecchi altri. Scrivere o telefonare a: Alberto Visani - Corso Garibaldi, 68 - 48018 Faenza (RA) - Tel. 0546/26462 28052

Vendo programmi per Spectrum 16-48 Kbyte. Per richiedere l'elenco gratuito, comprendente tutte le ultime novità, telefonare a: Massimo D'Ascenzo - Via Francesco D'Ovidio, 109 - 00137 Roma - Tel. 06/8280043

Vendo programmi di pronostici sulle "Corse di cavalli" per ZX Spectrum 16-48 Kbyte a L. 10.000. Massima serietà. Fabrizio Fulco - Piazzale del Fante, 49 - 90146 Palermo - Tel. 091/518089

Vendo, cambio programmi per ZX Spectrum. Dispongo di favolosi programmi come: Manic Miner, Alchemist, Atic Atac, Transword II. Per informazioni scrivere a: Franco Falcone - Via Deledda, 27 - 10024 Moncalieri (TO) - Tel. 011/6408088

Eccezionale! Vendo moltissimi meravigliosi programmi come Atic Atac, Manic Miner, Alchemist per ZX Spectrum 16-48 Kbyte. Prezzi a partire da L. 3.000 cadauno. Disponibili anche giochi BASIC e listati a richiesta. Per ricevere gratis l'elenco scrivere o telefonare ore pasti a: Domenico Crapanzano - Viale Regina Margherita - 98100 Messina - Tel. 090/56884

Vendo, cambio tutti i migliori programmi per ZX Spectrum 16-48 Kbyte: scrivete per ricevere gratuitamente il listino. Simone Frosini - Via Cesalpino, 20 - 52100 Arezzo

Cambio, vendo programmi per Spectrum 16-48 Kbyte. Spedite la vostra lista ed io vi invierò la mia. Se volete acquistare richiedermi la lista. Prezzi stracciati! Risposta assicurata. Gilberto Zeppi - Via Roma, 134 - 60030 Angeli di Rosora (AN) - Tel. 88010

Vendo programmi per tutti i computer Sinclair (ZX80, ZX81, Spectrum), per tutti i computer Commodore, Texas, Apple, Oric, HP, IBM PC, Olivetti, Sharp, Atari, DAI, PET, ecc, ecc, a prezzi imbattibili. Invio il catalogo a tutti coloro che invieranno L. 500. Francesco Corso - Via G. Cuomo, 53 - 84092 Bellizzi (SA)

Per ZX Spectrum 48 Kbyte vendo, cambio software di ottima qualità a prezzi eccezionali (L. 2.000 cad.). Scrivere o telefonare per invio lista a: Andrea Omiccioli - Via Principe Eugenio, 9 - 20455 Milano

Vendo, cambio programmi Spectrum 16-48 Kbyte. Vendo ogni programma a L. 1.500. Vendo raccolte che mi giungono ogni due mesi direttamente dall'Inghilterra. Per ogni ordine programmi in omaggio. E' possibile anche lo scambio. Massimiliano Nucci - Via Chimera, 22 - 52100 Arezzo - Tel. 0575/351195

Vendo, cambio software per Spectrum. Disponibili oltre 300 titoli per 16-48 Kbyte. Per informazioni scrivere o telefonare ore pasti. Prezzo inferiore alle L. 1.000. Antonio Manfredonio - Via Mergellina, 199 - 80122 Napoli - Tel. 081/684709

Cambio o vendo a L. 3.000 oltre 100 programmi per Spectrum con novità originali inglesi. Scrivetemi per la lista inviando la vostra. Serietà, convenienza, risposta sicura. G. Piero Santini - Via delle Viole, 3 - 60019 Senigallia (AN) - Tel. 071/659322

Per passaggio a sistema superiore vendo programmi per Spectrum 16-48 Kbyte disponibili su cassette C90 (minimo 22, massimo 45 programmi al prezzo di L. 12.000 ciascuna, pagamento contrassegno, spedizione compresa. Per avere l'elenco dei programmi spedire L. 500 in francobolli, nome, cognome ed indirizzo a: Giuseppe Castelli - Via Tolmino, 24 - 10141 Torino

Attenzione: amico (portatore di handi-cap) impossibilitato alla costruzione dei seguenti Kit, vende:
- corso completo teorico/pratico Scuola Radio Elettra per il montaggio di TV portatile b/n, Oscilloscopio 3", il tutto nuovo.

- Nuova Elettra Kit montati nuovi con contenitore; prova zener, prova hom altoparlanti, sonda wattmetro, wox con antiwox.

- Nuova Elettra Kit montati senza contenitore; temporizzatore tergi regolabile in battute e velocità, temporizz. tergi regolabile in battute, VFO 27/144 MHz.

- Materiale nuovo completo di contenitore: compressore microf. regolabile, filtro anti TVI 27 MHz, doppio portaradio estraibile auto/casa.

- Autocostruito (nuovo) lineare CB AM/SSB 50/100 watt transistor, autocostruito (nuovo) lineare CB AM/SSB 400 watt valvolare, regolabile in tre potenze.

- Consolle videogiochi a colori "Nicole", 28 giochi più joystick.

- Ricetrasmittente CB PAL/69 AM/SSB 69 CH mobile - completo di mik (usato ottimo).

- Ricetrasmittente CB Sommerc AM 23 CH mobile - completo di mik (usato ottimo).

- Televisore "Korting" 6 CH 23" b/n (usato ottimo).

- Riviste annate complete e non: "Elettr. pratica", "Elektor", "Cq eletr.", "Elettr. viva", "Break", "Onda quadra", "Selezione", "Radio kit". Davini Mauro - Via L. Da Vinci, 40 - 46019 Viadana (MN) - Tel. 0375/81384 (ore pasti).

Vendo ottime traduzioni in italiano dei migliori giochi per Spectrum: Hobbit L. 10.000 (20 pag.), Hurg L. 10.000, Voice Chess, Flight Simulation, Time Gate, Dictator, Piantina Atic Atac e Sabre Wolf L. 5.000 ognuno e molti altri ancora. Risposta in giornata. Ordinare o scrivere a: Mauro D'Orazi - Casella Postale 24 - 41012 Carpi (MO)

Vendo, cambio software ZX Spectrum 16-48 Kbyte. Programmi disponibili su cassetta: Xadom, The Pyramid, Road Racer, Ant Attack, Flight Simulation, Black Bomber, Meteor Storm e molti altri. Inviare la vostra lista dei programmi. A chiunque abbia il programma Back-up (per sprotteggere i programmi) e me ne fa avere una copia cedo gratis 5 programmi a sua scelta. Angelo Presti - Via Vespri, 2 - 93012 Gela (CL) - Tel. 0933/912677 912763

SUPERSING

CECOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.soft - Viale Restelli, 5 - 20124 Milano - Tel. 6986228

LIBRI

The complete Spectrum (Lo Spectrum completo)
Cod. BGRC015 L. 26.000
L'hardware dello Spectrum - i componenti del Sinclair Spectrum e come
intervenire nella sua struttura
Cod. AMUD025 L. 12.000

Ordino i seguenti libri per un importo totale di L. + L. 2.000
come contributo fisso per spese di spedizione

- Cod. Cod.
Cod. Cod.
 Contanti allegati Assegno allegato n°
-
 Ho spedito l'importo a mezzo vaglia postale
 Ho versato l'importo sul cc. n° 19445204 intestato a J.soft - Milano
 Pagherò in contassegno al postino al ricevimento dei volumi (valido solo per i
soci in Italia)

Nome
Cognome
Via
Città C.A.P.
Se richiesta fattura:
Cod. F. e P. Iva
Data
Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSING INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a
tutti i lettori. Chi desidera contribuire, vendere o cam-
biare hardware o software può farlo rivolgendosi
a J.soft - Viale Restelli, 5 - 20124 MILANO

COMPRO VENDO CAMBIO
 ZX80 ZX81 ZX Spectrum QL
 PERIF. SOFTWARE

.....
.....
.....
Nome Cognome
Via C.A.P.
Città Tel.

**SUPERSING è bello, però... (ovvero suggerimenti,
idee, critiche, richieste e tutto ciò che vi passa per
la testa).**

.....
.....
.....
Nome Cognome
Via C.A.P.
Città Tel.

Dietro un bambino maltrattato c'è sempre una famiglia in crisi.

È impossibile stabilire con esattezza quanti bambini in Italia subiscono violenze. Tanti, purtroppo. Ma una cosa è certa: la violenza che oggi un bambino subisce e impara, da grande la ritorcerà contro la società. Bisogna intervenire, e subito: agendo sulle cause. Intervenedo sulla crisi familiare che è sempre alla base dei maltrattamenti a un bambino.

**"Vorrei contribuire anch'io".
Per sostenere il C.A.F. vi invio**

- L. 10.000 L. 25.000 L. 50.000
 L. 100.000 L. 500.000 L.

- Tramite assegno bancario intestato C.A.F.
Via V. E. Orlando 15, 20142 Milano, che vi invio,
unito a questo tagliando.
 Tramite versamento sul C.C.P. n° 22349203 intestato C.A.F.
Via V. E. Orlando 15, 20142 Milano.

Nome e Cognome

Indirizzo

CAP..... Città

Provincia

Attendo: ricevuta e tesserino di "Sostenitore" del C.A.F.

JAR

Segnaliamo ai Servizi Sociali del Territorio i casi di maltrattamento.

La prima cosa da fare: segnalare ogni caso di maltrattamento di cui siamo a conoscenza o di cui abbiamo il fondato sospetto. Telefonate ai Servizi Sociali del Territorio dal lunedì al venerdì,

dalle ore 9 alle ore 12 (troverete i numeri di telefono su Tuttocittà), oppure al Tribunale per i Minorenni (chiedendo del giudice di turno) dal lunedì al sabato dalle ore 9 alle ore 13, telefono 4988001/2/3: oppure al C.A.F., a questi numeri di telefono 02/8265051/2, che rispondono 24 ore su 24, 7 giorni su 7.

Sosteniamo il C.A.F.

Il C.A.F. è il primo Centro di Aiuto al Bambino Maltrattato e alla Famiglia in Crisi, sorto in Italia con lo scopo di intervenire nei casi di maltrattamento.

Oggi, nella sua nuova sede più grande e attrezzata, accoglie provvisoriamente i bambini maltrattati, contemporaneamente svolge un'azione terapeutica sulla famiglia per ripristinare l'equilibrio e reintegrare il bambino nel suo ambiente.

Il C.A.F. è un'istituzione privata che opera in collaborazione con i Servizi Sociali, con i supporti messi gratuitamente a disposizione dal Comune di Milano, e con i contributi finanziari volontari dei cittadini.

Contribuisci anche tu, utilizzando il tagliando. È un gesto necessario, per i bambini maltrattati, per te che vuoi fare qualcosa di concreto contro la violenza.

Spazio gratuito offerto da J.soft

**UN BAMBINO
MALTRATTATO OGGI
SARÀ UN UOMO
VIOLENTO DOMANI.
FACCIAMO QUALCOSA
PRIMA.**

C.A.F.



**CENTRO DI AIUTO
AL BAMBINO MALTRATTATO
E ALLA FAMIGLIA IN CRISI**

Dalla grande edicola Jackson

Tutto sull'hobby e home computer



STRUMENTI MUSICALI

In questo numero:
**Speciale equalizzatori
 Gibson Explorer
 Los Angeles guitar show
 Novità Francoforte '85**



VIDEOGIOCHI

In questo numero:
**Las Vegas '85
 Designer's pencil
 15 giochi-novità
 I Video game fai-da-te**



HOME COMPUTER

In questo numero:
**Spectravideo
 728 MSX
 Ghostbuster
 per CBM 64
 Un adventure
 per il QL
 Match day
 per Spectrum
 C 16 contro VIC 20**

Strumenti Musicali/Video Giochi/Home Computer
 sono pubblicazioni firmate:



GRUPPO EDITORIALE JACKSON

via Rosellini, 12-20124 Milano